



Addie water softener installation instructions

[{ type: thumbs down, id: missingTheInformationINeed, label:Missing information I need }, { type: thumbs down, id: tooComplicatedTooManySteps, label:Out of date }, { type: thumb down, id: samplesCodeIssue, label:Samples/Code issue }, { type: thumb-down, id: otherDown, label:Other }] [type: thumbs up, id: easyToUnderstand, label:Easy to understand },{ type: thumb-up, id: solvedMyProblem, label:Solved my problem }, id: solvedMyProblem, label:Solved my problem }, type: thumb-up, id: solvedMyProblem, label:Solved my problem }, type: thum proxying a Web service through a server when you use an API in a mobile app to protect your API key. With directions API, you can search for directions using a series of waypoints. Specify the origin, destination and point of the path as text strings (e.g. Chicago, IL or Darwin, NT, Australia), as place IDs or as latitude/longitude coordinates API returns the most efficient routes when calculating directions. Travel time is the primary optimized factor, but the API can take into account other factors such as distance, number of turns and more when deciding which route is most effective. Keep on the upside: This service is not designed to respond in real time to user input. For dynamic instructions calculations (for example, within a user interface element), consult the JavaScript API Direction data within maps provided by one of the Google Maps APIs. Provides an introduction to the use of APIs and reference material about available parameters. Before you start developing with the Directions API, review the authentication requirements (you must enable billing on your project). Instructions to ask for Instructions API request has the following format: where outputFormat can be any of the following values: json (recommended) denotes output in JavaScript Object Notation (JSON) xml denotes output as XML Note: URLs must be properly encoded to be valid and are limited to 8192 characters for all web services. Be aware of this limit when creating URLs. HTTPS or HTTP Security is important and HTTPS is recommended whenever possible, especially for apps that include sensitive user data, such as the user's in the requirements. Using HTTPS encryption makes your app safer and more resistant to snooping or tampering. If HTTPS is not possible, use: Required parameters are optional. As is the standard in URLs, all parameters are separated by the ampersand (&) character. All reserved characters (for example, the plus +sign) must be coded with a URL. The list of parameters and their possible values are listed below. Origin of the required parameters – Location ID, address, or text value of latitude/latitude from which you want to calculate the instructions. The set IDs must be prefixed place_id:. You can retrieve site IDs from the Geocoding API and API sites (including Place Autocomplete). For example, using place autocomplete place IDs, see Place Autocomplete and Directions. For more information about location IDs, see the site ID overview. origin=place_id:ChIJ3S-JXmauEmsRUclaWtf4MzE If you pass the address, directions geocodings the string and converts it to latitude/longitude coordinate to calculate the instructions. This coordinate may be different from the one returned by the GEOCODING API, for example the entrance to the building and not to its center. origin=24+Sussex+Drive+Ottawa+ON Note: using a location ID is preferred instead of using addresses or latitude/longitude coordinates. Using a location ID is preferred instead of using addresses or latitude/longitude coordinates. not be an access point to the facility, or even a road that will quickly or safely lead to the destination. If you pass the coordinates, the point will be swing to the nearest road. Adding a location ID is preferred. If you pass the coordinates, the point will be swing to the nearest road that will quickly or safely lead to the destination. If you pass the coordinates, the point will be swing to the nearest road. or stacked code. Format plus codes as shown here (plus characters are url-fled to %2B and spaces are URL-escaped to %20): the global code is 6 character or longer local code with explicit location (CWC8 + R9 Mountain View, CA, USA is CWC8%2BR9%20Mountain%20View%20CA%20USA). destination — Location ID, address, or text value of latitude/latitude to which you want to calculate the instructions. The options for the destination parameter are the same as for the origin parameter described above. Key – Your app's API key. This key identifies your application for quota management purposes. Learn how to get the key. For example, please note: Users of the Google Maps Platform Premium Plan can use either a API key or a valid client ID and a digital signature in your directions requirements. More about authentication parameters, see Premium plan users. Plan. parameter mode (driving default) — Specifies the mode of transport to be used when calculating the instructions. Valid values and other application details are listed in travel methods. waypoints — Lists a number of intermous locations to include along the route by directing it through a specific location(s). The API supports points of travel for these modes of travel: driving, walking and cycling; not transit. You can specify path points using the following values: Set ID: A unique location-specific value. This value is available only if the request includes the Google Maps Platform Premium Plan client API key or ID (ChIJGwVKWe5w444kRcr4b9E25-Go. Address line (Charlestown, Boston, MA) Latitude/Latitude Coordinates(lat/lng): an explicit pair of values. (-34.92788%2C138.60008 comma, no space) Kodio polylin which can be determined by a set of any of the above. (enc:lexeF{~wsZejrPjtye@:) For examples and more information about trip points, see the trip point, see the trip point guide below. Alternatives – If set to true, it states that the route service can provide more than one alternative route in response. Keep at the beginning that providing alternatives to the route can increase response time from the server. This is only available for requests without an interse point. avoid — Specifies that the calculated route should avoid toll roads/bridges. motorways indicate that the calculated route should avoid motorways. ferries indicate that ferries should be avoided by the calculated route. indoors states that the calculated route estrictions below. language - The language by which results are returned. See a list of supported languages. Google frequently updates supported language is not delivered, the API tries to use the preferred language is not delivered, the API does its best to provide a street addresses that is readable for both the user and the local population. To achieve this goal, it returns street addresses are returned in the preferred language. All other addresses are returned in the same language as selected from the first component. If the name is not available on the language, the API uses the nearest match. Preferred has little impact on the set of results that the API decides to return and the order in which they return. The geocoder interprets abbreviations differently depending on the language, such as abbreviations for street types or synonyms that may be valid in one language, but not in another. For example, utca and tér are synonymous with the street in Hungarian. unit — Specified as ccTLD (top-level domain) value of two characters. (For more information, see the region bias below.) arrival_time — Specifies the desired arrival time for transit routes, in seconds from midnight, 1 January 1970 UTC. You can determine either departure_time or arrival_time, but not both. Keep in arrival_time that these rules must be stated as integer in seconds from midnight, January 1, 1970 UTC. UTC. If departure_time 9999-12-31T23:59:59:59:999999999, the API will return departure at the current time (right at the nearest second). The departure time can be stated in two cases: For requirements where the mode of travel is in transit: you can optionally specify one of the departure_time or arrival_time. If no time is specified, departure_time to receive the route and duration of the trip (response field: duration_in_traffic) that take into account traffic conditions. This option is available only if the request contains a valid API key or a valid Google Maps Platform Premium Plan client ID and signature. Departure time is not specified, the route choice and duration are based on the road network and average time-independent traffic conditions. The results for the default request may vary over time due to changes in the road network, updated average traffic conditions and the distributed nature of the service. Results can also vary between almost equivalent routes at any time or frequency. traffic_model (default settings best_guess) – specifies the assumptions to be used when calculating traffic times. This setting affects the value returned to the duration_in_traffic in the reply, which contains the estimated traffic time based on historical averages. The traffic_model parameter can only be specified for driving directions where the request includes an API key or Google Maps Client ID premium plan. The available values for this parameter are: best_guess (default) indicates that the returned duration_in_traffic should be the best estimate of travel time given what is known about both historical traffic should be longer than the actual travel time on most days, although occasional days with particularly poor traffic conditions can exceed that value. optimistically indicates that the duration_in_traffic should be shorter than the actual travel time on most days, although occasional days with particularly good traffic conditions may be faster than thet. The default value will best_guess the most useful predictions for the vast majority of usage cases. It's possible that best_guess travel time to be shorter than optimistic, or alternatively, longer than pessimistic, because of the way the prediction model best_guess integrates live traffic information. transit_mode — Specifies one or more preferred modes of transport. This parameter can only be specified for public transport instructions, and only if the request includes a Google Maps Platform Premium Plan API key or ID. The parameter supports the following arguments: the bus indicates that the calculated route should prefer bus travel. The subway indicates that the calculated route should prefer train indicates that the calculated route should prefer bus travel. The subway indicates that the calculated route should prefer bus travel. route should prefer travel by train, tram, light rail and metro. This is the equivalent of transit_mode=train|tram|subway. transit_routing_preference — Specifies preferences for transit routes. Using this parameter, you can biasedly restore options, instead of accepting the default best route chosen by the API. This parameter can only be specified for public transport instructions, and only if the request includes a Google Maps Platform Premium Plan API key or ID. The parameter supports the following arguments: less_walking indicates that the calculated route should prefer a limited number of transfers. An example of the instruction prompted the next request returns driving directions from Toronto, Ontario to Montreal, Quebec. what? origin=Toronto&destination=Montreal & apoiding parameters, the initial request can be modified to return routes for scenic bike journeys that avoid major highways. what? origin=Toronto&destination=Montreal The following request seeks transit instructions from Brooklyn, New York to Queens, New York. The request does not specify departure_time, so the departure_time is set at the current time: origin=Brooklyn&destination=Queens&mode=transit&key=YOUR_API_KEY The next transit request includes a specific departure time. Please note: In this example, the departure time is specified as July 30, 2012 at 09:45 hours. To avoid an error, you must change the parameter at a time in the future before submitting a request. what? origin=Brooklyn&destination=Queens & amp;departure_time=1343641500&mode=transit&destination=Queens & amp;destination=Queens & amp;destination=Qu origin=H8MW%2BWP%20Kolkata%20India&destination=GCG2%2B3M%20Kolkata%20India&key=YOUR API KEY The following request returns driving directions from Glasgow, UK to Perth in the UK using the place ID. what? origin=place id:ChIJ685WIFYViEgRHIHvBbiD5nE & amp;destination=place id:ChIJA01I-8YVhkgRGJb0fW4UX7Y &key=YOUR_API_KEY Travel mode When you calculate the instructions, you can specify a mode of transport to use. By default, the instructions for driving using the road network. walking requires walking directions to footpaths and sidewalks (where available). cycling requires cycling directions via bike paths and preferred streets (where available). If you set up a mode of transport, you can specify either departure time is set to the current time). You can also optionally transit mode and/or transit routing preference. Keep it at the beginning: both pedestrian and cycling routes sometimes can't include clear walking or bike paths, so these directions will return the warnings in the returned result that you must show to the user. Traffic Information Caution: Requests using traffic information are charged at a higher rate. Learn more about how to charge api using instructions. Traffic information is used when all of the following are applied (these are the conditions required to receive the duration_in_traffic in the response: The travel mode parameter is or is not specified (driving is the default travel method). The request includes a valid parameter departure_time. The text departure_time be set to the current time or for some time in the future. It can't be in the past. The request does not If the request includes points of the trip; to affect the route, but avoid stopping. For example, & amp;waypoints=via:San Francisco|via:Mountain View]... Optionally, you can traffic_model in your request the assumptions you will use when calculating traffic times. The following URL triggers a request to travel from Boston, MA to Concord, MA, via Charlestown and Lexington. The request shall include the time of departure, the fulfilment of all conditions for returning the duration_in_traffic in response directions. what? origin=Boston,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&destination=Concord,MA&dest & amp;key=YOUR_API_KEY Waypoints Caution: Requirements that use more than 10 points (between 11 and 25) or path optimization, are charged at a higher speed. Learn more about google maps product billing. When calculating routes using the API instructions, you can specify the path points to return a route that includes passing or stopping in middle locations. Driving, walking or cycling, but not driving directions, but not in transit routes. Specify the locations in the path point parameter. You can deliver one or more locations calculates the route using the intermediate in the order they are given. The advantage of parsing a point value is the location ID, the latitude/longitude coordinates, and then the address. If you pass the site ID, you must prefix it place_id: and API sites (including Place Autocomplete). For example, using place autocomplete place Autocomplete and Directions. For more information about location IDs, see the site ID overview. For efficiency and accuracy, use location IDs when possible. These IDs are uniquely explicit like lat/Ing value pairs and traffic variables. Unlike an address, IDs do not require a search service or an intersex request for location details; therefore the performance is better. If you pass latitude/longitude coordinates, the values go directly to the front server to calculate the instructions without geocoding. The dots are cracked on the roads and may not provide the accuracy your app needs. Use coordinates when you're sure that the values actually determine the points your app needs to be routed, regardless of possible access points or additional geocode details. Ensure that the comma (%2C) rather than the space (%20) separates the geographical values and length. If you pass the address value is ambiguous, the value could trigger a search to parse from similar addresses. For example, street 1 may be full value or partial value for 1. This result may be different from the one returned by the Geocoding API. You can deliver a codided set of dots using a kodided polylin algorithm. You will see that the kodided set is useful for a large number of dots, because the URL is significantly shorter. All web services have a URL limit of 8192 characters. Encom coded polylines, separated by the pipe sign (]). For example, waypoints=enc:gfo}EtohhU:. You can also turn on multiple co-oid polylines, separated by the pipe sign (]). For example, waypoints=enc:gfo}EtohhU:. You can also turn on multiple co-oid polylines, separated by the pipe sign (]). via:enc:udymA{~bxM: The following URL triggers a request to travel between Boston, MA and Concord, MA with stops in Charlestown and Lexington, in that order. Example uses addresses: origin=Boston, MA&waypoints=Charlestown, MA| Lexington, in that order. point of the trip in the request, the prompt prompt adds entry down the leg to provide details for stops on that section of the trip. If you want to affect your route using stop points with the path: will not add entry to the down the leg, but will route the journey through the point of the trip. The following URL changes the previous request so that the trip is routed through Lexington without stopping: origin=Boston,MA&waypoints=Charlestown,MA|via:Lexington,MA&waypoints=Charlestown,MA|via:Lexington,MA&waypoints=Charlestown,MA|via:Lexington,MA&waypoints=Charlestown,MA|via:Lexington,MA&waypoints=Charlestown,MA|via:Lexington,MA&waypoints=Charlestown,MA|via:Lexington,MA&waypoints=Charlestown,MA|via:Lexington,MA&waypoints=Charlestown,MA|via:Lexington,MA&waypoints=Charlestown,MA|via:Lexington,MA&waypoints=Charlestown,MA|via:Lexington,MA&waypoints=Charlestown,MA|via:Lexington,MA&waypoints=Charlestown,MA|via:Lexington,MA&waypoints=Charlestown,MA|via:Lexington,MA&waypoints=Charlestown,MA|via:Lexington,MA&waypoints=Charlestown,MA|via:Lexington,MA&waypoints=Charlestown,MA|via:Lexington,MA&waypoints=Charlestown,MA|via:Lexington,MA&waypoints=Charlestown,MA|via:Lexington,MA&waypoints=Charlestown,MA|via:Lexington,MA&waypoints=Charlestown,MA|via:Lexington,MA&waypoints=Charlestown,MA|via:Lexington,MA&waypoints=Charlestown,MA|via:Lexington,MA&waypoints=Charlestown,MA|via:Lexington,MA&waypoints=Charlestown,MA|via:Lexington,MA&waypoints=Charlestown,MA|via:Lexington,MA&waypoints=Charlestown,MA|via:Lexington,MA&waypoints=Charlestown,MA|via:Lexington,MA&waypoints=Charlestown,MA|via:Lexington,MA&waypoints=Charlestown,MA|via:Lexington,MA&waypoints=Charlestown,MA|via:Lexington,MA&waypoints=Charlestown,MA|via:Lexington,MA&waypoints=Charlestown,MA|via:Lexington,MA&waypoints=Charlestown,MA|via:Lexington,MA&waypoints=Charlestown,MA|via:Lexington,MA&waypoints=Charlestown,MA|via:Lexington,MA&waypoints=Charlestown,MA|via:Lexington,MA&waypoints=Charlestown,MA|via:Lexington might look like in real time and helps ensure that the path points are located in locations that are available with the Directions API. Caution: Using the route: prefix to avoid stopping results in directions that are strict in their interpretation of the point of the way. This interpretation may result in serious detours on the route or ZERO RESULTS in the response status code if API directions cannot create instructions through that point. The following URL searches the points of the path using latitude/longitude coordinates: origin=sydney,au&destination=perth,au&waypoints=via:enc:lexeF{~wsZejrPjtye@: & amp;key=YOUR_API_KEY Optimize your points by default, Directions calculates the route through the predicted times in the default order. (This optimization is the application of a travel salesman problem.) Travel time is the primary factor that is optimized, but other factors such as distance, number of bends and more can be taken into account when deciding which route is most effective. All points must be stopped for service directions in order to optimize your route. If you order Directions to optimize the order of your dots, their order will be returned to the waypoint order field within the route object. The new waypoint order field returns non-zero-based values. The following example calculates the road journey from Adelaide, SA & amp; waypoints=optimism: true Barossa+Valley, SA Clare, SA| Connawarra, SA| McLaren+Vale, SA & amp;key=YOUR_API_KEY The calculated route overview will indicate that the calculated route overview will indicate that the calculated at a higher rate. Learn more about how to charge for Google Platform products. Instructions for restrictions that adhere to certain restrictions may be calculated. The restrictions are indicated by the use of avoidance parameters and an argument on that parameters are indicated by the use of avoidance parameters and an argument on that parameters are indicated by the use of avoidance parameters and an argument on that parameters are indicated by the use of avoidance parameters are indicated by the use of a passing both restrictions to the avoidance parameter. For example: avoid=tolls|highways|ferries. Keep at the beginning: Adding restrictions for unit systems contain text within the distance field that can be displayed to the user to indicate the distance of a specific route step. By default, this text uses the country or region of origin unit system. For example, a route from Chicago, IL to Toronto, ONT will show results in miles. You can override this unit system by explicitly placing it within the unit parameter passing one the following values: The metric determines the use of the measuring system. Text distances return to miles and meters. imperial specifies the use of the imperial (English) system. Text distance field. Distance fields also contain values that are always expressed in meters. Region bias You can set a routes service to return results from a specific region by using the region parameter. This parameter takes ccTLD (the top-level country code domain) that specifies the region's bias. Most ccTLD is uk (.co.uk) while its ISO 3166-1 codes are identical to ISO 3166-1 codes, with some notable exceptions. For example, the UK ccTLD is uk (.co.uk) while its ISO 3166-1 codes are identical to ISO 3166-1 use any domain where the main Google Maps app has run driving directions. For example, the request for direction for Toledo to Madrid returns the corresponding results when the region is set to es, and Toledo is then interpreted as a Spanish city: origin=Toledo&destination=Madrid®ion=es&key=YOUR_API_KEY { status: OK, routes: [{ summary: AP-41, legs: [{ } copyrights: Map data ©2010 Europa Technologies, Tele Atlas, warnings: [], waypoint_order: [] } A directions request for Toledo to Madrid sent without parameters of the region does not return results, because Toledo is interpreted as a city in Ohio and not in Spain: origin=Toledo&key=YOUR_API_KEY { status: ZERO_RESULTS, routes: [] } Directions responses Directions are returned in the format indicated by the output flag within the request URL path. Sample responses Below shows a sample of HTTP requests, calculating routes from Chicago, IL to Los Angeles, CA via two-point routes in Joplin, MO and Oklahoma City, OK. what? origin=Chicago, IL& & amp;waypoints=Joplin,MO| Oklahoma+City,OK & amp;key=YOUR API KEY The above example looks for a JSON output. You can also request an XML output. Click the tabs below to see the JSON and XML responses on the sample. Since the results of the directions can be quite extensive, the repeated elements within the response are omitted for clarity. JSON { status: OK, geocoder status : OK, place id : ChIJ7cv00DwsDogRAMDACa2m4K8, types : [locality, political] } { geocoder status : OK, place id : ChIJ69Pk6jdlyIcRDgM1KDY3Fpg, types : [locality, political] } { geocoder status : OK, place id : ChIJ69Pk6jdlyIcRDgM1KDY3Fpg, types : [locality, political] } { geocoder status : OK, place id : ChIJ69Pk6jdlyIcRDgM1KDY3Fpg, types : [locality, political] } : [lokalitet, politički]]], rute: [{ sažetak: I-40 W, noge: [{ koraci: [{ travel_mode: VOŽNJA, start_location: { lat: 41.8507300, lng: -87.6514100 }, polilin: { bodovi: a~l~ Fjk~ uOwHJy@P }, trajanje: { vrijednost: 19, tekst: 1 min }, html_instructions: Head \u003cb\u003chu003chu003chu003cb\u003cb\u003cb\u003cb\u003cb\u003cb\u003cb\u003cb\u003cb} St\\u003c/b\u003e prema \u003cb\u003e prema \u003cb\u003eW Čermak Rd\u003c/b\u003e, udaljenost: { vrijednost: 2137146, tekst: 0,1 mi } , udaljenost: { vrijednost: 2137146, tekst: 1.328 mi }, start_location: { lat: 35.4675602, lng: -97.5164276 }, end_location: { lat: 34.0522342, lng: -118.2436849 } start_address: Oklahoma City, OK, SAD, end_address: Los Angeles, CA, USA }], copyrights: Map data ©2010 Google, Sanborn, overview_polyline. { bodovi: a~ ~ Fjk ~ uOnzh@vlbBtc ~ @tsE'vnApw {A'dw@ ~ w \\[tNtqf@l {Yd_Fblh@rxo@b} @xxSfytA blk@xxaBeJxlcBb ~ t@zbh@jc| Bx}C'rv@rw|@rlhA~ dVzeo@vrSnc}Axf]fjz@xfFbw~ @dz{A~ d{A|zOxbrBbdUvpo@' cFp~ xBc'Hk@nurDznmFfwMbwz@bbl@lq~@loPpxq@bw @v|{ CbtY ~ jGqeMb {iF|n\\~ mbDzeVh Wr| Efc\\x'lj{kE}mAb ~uF{cNd}xBjp]fulBiwJpqq@|kHntyArpb@bijCk Kv~ eGyqTj |@'uV'k| DcsNdwxAott@r}q@ gc@nu'CnvHx'k@dse @j|p@zpiAp|qEicy@'omFvaErfo@igQxnlApqGze~ AsyRzrjAb @ftyB}pllo BflmA~yQftNboWzoAlzp@mz'@|} Imajte na naci da ove rezultate općenito treba raščlanjujući JSON je relativno lako. Pogledajte <type> <place_id>ChIJ7cv00DwsDogRAMDACa2m4K8</place_id> <type> <type& <geocoded_waypoint> <geocoder_status>OK</geocoder_status> <type> <type> <type> <type> <type> <geocoded_waypoint> <geo ChIJE9on3F3HwoAR9AhGJW fL-l</place id> </route> </leg> </leg> </start location> </start location> </leg> </leg> </leg> </start location> </leg> </start location> </start l <end_location> <lat>41.8525800</lat> <lng> 8lt;lng> 8lt;/end_location> </end_location> </end_ Cermak Rd <distance> <value> 207</value> 207</value> <value> <lat>35.4675602</lat> <lat>35.4675602</lat> <lat> SAD</end_address> Podaci o <copyrights>karti ©2010 Google , Sanborn</copyrights> <overview_polyline> <points>a ~ I ~ Fjk ~ uOnzh@vlbBtc ~ @tsE'vnApw {A'dw@ ~ w \tNtqf@I {Yd_Fblh@xxaBeJ xlcBb ~ t@zbh@jc] Bx}C'rv@rw|@rlhA~ dVzeo@vrSnc}Axf]fjz@xfFbw~ @dz{A~ d{A|zOxbrBbdUvpo@'cFp~ xBc'Hk@nur DznmFfwMbwz@bbl@lq~@loPpxq@bw_@v|{ CbtY ~ jGqeMb {iF|n \~ mbDzeVh_Wr| Efc\x'lj{kE}mAb~uF{cNd}xBjp]fulBiw Jpgg@|kHntyArpb@bijCk_Kv~ eGyqTj_|@'uV'k| DcsNdwxAott@r}q@_gc@nu'CnvHx'k@dse@j|p@zpiAp|gEicy@'omFva Erfo@igQxnlApqGze~AsyRzrjAb__@ftyB}pllo_BflmA~yQftNboWzoAlzp@mz'@|} _@fda@jakEitAn{fB a]lexClshBtm qAdmY hLxiZd~ XtaBndgC</points> </overview_polyline> <lat>34,0523600</lat> <lat>41.8781100</lat> <lat>34,0523600</lat> <lat>41.8781100</lat> <lat>34,0523600</lat> </northeast> </northeast> <lat>34,0523600</lat> </northeast> <lat>34,0523600</lat> </northeast> </northeast> </northeast> </northeast> </northeast> </ng> </northeast> </ng> </northeast> < </bounds> natuknica da se odgovor XML-a sastoji od jednog i <DirectionsResponse>sljedećih elemenata najviše razine: sadrži <status>metapodatke o zahtjevu. Pogledajte kodove statusa u nastavku. Jedan <geocoded waypoint>po međutosu, plus podrijetlo i odredište, s pojedinostima o rezultatu geokodinga. Možda postoje prazni <geocoded waypoint></geocoded waypoint> elementi. Pogledajte Geokodidne točke puta u nastavku. Nula ili više <route>elemenata, od kojih svaki sadrži jedan skup informacija o usmjeravanju između podrijetla i odredišta. Preporučujemo use json as your preferred output flag, unless your service requires xml for some reason. Processing XML trees requires a certain care, so you refer to the appropriate cumins and elements. See Parsing XML with XPath for some recommended design forms for processing outputs. The rest of this documentation will use the JSON syntax. In most cases, the output format is not important for the purpose of illustrating the concepts or field names in the documentation. However, note at the beginning the following subtle differences: the XML results are wrapped in the </DirectionsResponse> <//DirectionsResponse> <//Dire <step>are <leg>i). JSON denotes the order of the paths by the waypoint_order fields, while XML marks them using individual <waypoint_index>elements. Blank elements are marked through blank strings in JSON, but by the absence of any such element in XML. A response that does not generate results will return an empty network of routes in JSON, but by the absence of any such element in XML. <route>without elements in XML, for example. Response instructions The answers contain the following root elements: the status contains a series detailing the geocode of origin, destination and trip points. See the Geocoded points of the path below. routes contain a number of routes from origin to destination. See Routes below. Routes consist of nested legs and steps. available_travel_modes contains the available travel methods in the countries of a specific set of points. This field is not returned if one or more points are by: intermediate. See below for details. The status codes of the status field within the instruction response object contain the status field can contain the following values: OK indicates that the response contains a valid result. NOT FOUND shows that at least one of the locations listed in the origin, destination or exchange of applications could not have been geocodisedt, ZERO RESULTS shows that there are too many points listed in the applications that use the Directions API as a web service or instruction service in the Map JavaScript API, the maximum number of points allowed is 25, plus origin and destination. MAX ROUTE LENGTH EXCEEDED shows that the requested route is too long and cannot be processed. This error occurs when more complex instructions are returned. Try reducing the number of points, turns, or directions. INVALID REQUEST states that the application submitted was invalid. Common causes of this status include an invalid parameter or parameter or parameter or parameter or parameter value. OVER DAILY LIMIT indicates any of the following: The API key is missing or invalid. Billing is not enabled in your account. The self-imposed cap for use has been exceeded. The scheduled payment method is no longer valid (for example, the credit card has expired). See the frequently set map rules to learn how to fix it. OVER_QUERY_LIMIT shows that the service was declined</route> </waypoint_index> </leg> instructions upon your application. UNKNOWN_ERROR indicates that the request for directions cannot be processed because of a server error. The request may fail if you try again. Error message field within the object to respond directions. This field contains more detailed information about the reasons for the specified status code. Keep on the upside: This field is not guaranteed to always be present, and its content is subject to change. Geocode points Geocoded waypoints series. They can be used to determine why the service would return unexpected or no routes. The elements in the geocoded waypoints series. series correspond, in their zero position, origin, to the points of order specified and the destination. Each element includes the following details about the geocoded operation for the corresponding point of the trip: geocoder_status indicates that no errors have occurred; the address was successfully broken down and at least one geocode was restored. ZERO_RESULTS suggests the geocode was successful, but did not return the results. This can happen if a non-existent address has been forwarded to the geocode was successful, but did not return the results.

the requested address. You may want to examine the original request for a misspelled and/or incomplete address. Partial matches can also be returned when the request corresponds to two or more locations in the same locality. For example, 21 Henr St, Bristol, UK will return a partial match for both Henry Street and Henrietta Street. Note, for example, that if the request includes a misspelled address. Suggestions run this way will also be marked as a partial match. place id is a unique identifier that can be used with another Google API. For example, you can use place id from the Google Place autocomplete response to calculate instructions. The following types are returned: street address indicates the precise address of the street. route indicates a named route (such as US 101). the intersection points to a large intersection, usually the two main roads. political entity. Usually, this guy points to a civilian administration training ground. national political entity and is usually the highest order of returned the geocoder. administrative level. In most cases, administrative_area_level_1 short names will closely coincide with ISO 3166-2 divisions and other widespread lists; however, this is not guaranteed because our geocoded results are based on different signals and location data. administrative levels are counties. Not all nations show this administrative level. administrative_area_level_3 points to a smaller civic department. Not all nations show this administrative_area_level_4 points to a smaller civic department. Not all nations area_level_5 points to a smaller civic department. Not all nations show this administrative level. colloquial_area indicates the most commonly used alternative name for the subject. locality is indicated on a built-in city or city political entity. the sublocality means the civic entity of the first row below the locality. For some locations, you can get one of the additional types: sublocality level 1 on sublocality level 5. Every level of sublocality level 1 on sublocality level 5. Every level of sublocality level 5. Every level 5. Every level of sublocality level 5. Every level of sublocality level 5. Every level 5. Every level of sublocality level 5. Every level 5. Ev with a common name of sub-plating indicates the subject of the first row below the named location, usually a unique buildings with a common name plus_code indicates a codeided location reference, derived from latitude and longitude. Plus codes can be used as a substitute for street addresses in places where they don't exist (where buildings are not counted or the street is not named). See for details in the . postal_code lists the postcode used to address mail within the country. natural_feature points to a designated point of interest. Typically, these POIs are prominent local entities that do not easily fit into another category, such as the Empire State Building or the Eiffel Tower. A blank list of types indicates that there are no known types for a specific address component, for example, Lieu-dit in France. These details will not be present for points listed as latitude/latitude text values if the service does not return such points are geocoded only to obtain their representative address after the route has been found. The empty JSON object will take up the appropriate seats in the geocoded_waypoints string. Even if the origin and/or destination does not exist), it still returns an empty route string. (XML replies consist of zero or more <route>elements.) Each element of the route field contains one result of the specified origin and destination. This route may consist of one or more legs depending on whether any points are listed. The route field can contain the following fields: the summary contains a string contains a brief text description of the route, suitable for naming and deploying routes from alternatives. [] contains a string contain the following fields: the summary contains a string contain the following fields: the summary contains a string contains a string contain the following fields: the summary contains a string contain the following fields: the summary contains a string contains a string contains within a specific route. A separate leg will be present for each point or destination specified. (The route without a cross-section will contain exactly one leg inside a string of legs.) Each leg consists of a series of steps. (See leg instructions below.) waypoint_order (or <waypoint_index>in XML) contains a string that indicates the order of any points in the calculated route. These points can be oversized if the request is forwarded to optimize:true within the parameter of its intermediates. overview_polyline contains one dotted object containing a coded polylin route view. This polylin is an approximate (smoothed) path of the resulting directions. the boundaries contain a vista of a box of overview_polyline. copyright text to be displayed for this route. You must handle and display these warnings yourself. fare: If present, it contains the total price (that is, the total cost of tickets) on this route. This property is returned only for transit legs is available. Information includes: currency: ISO 4217 currency code that indicates the currency to be valued: Total price amount, in the currency listed above. text: Total price amount, formatted in the language requested. For example, please note: API directions returns only pricing information for requests that contain either a customer API key or a client ID and a digital signature. Learn how to get to the key or how to use authentication parameters for Premium Plan users. Below is an example of information about prices within the route: routes: [{ borders</waypoint_index> & lt;/route> { northeast : { lat : 37.8079996, LNG : -122.4074334 }, southwest : { lat : 37.7881005, lng : -122.4203553 } }, copyrights : Map data ©2015 Google, fare : { currency : USD, value : 6 text : \$6.00 } Legs Each element in a row of legs determines one leg part of the journey from origin to destination in the calculated route. For routes that do not contain points of travel, the route will consist of one leg, but for routes that define one or more legs, which corresponds to the specific legs of the journey. Each leg within the leg field can contain the following fields: the steps[] contain a series of steps that indicate information about each separate step of the leg of the journey. (See the steps below.) distance indicates the total distance covered by this foot, as a field with the following elements: the value indicates the distance display, shown in units used at origin (or as overridden within the unit parameter in the request). (For example, miles and feet will be used for any origin within the United States.) Note that no matter which unit system is displayed as text, the distance field value always contains a humanreadable duration view. These fields may be missed if the duration is unknown. duration is returned of traffic conditions. See traffic model for options you can use to make sure the value returned is optimistic, pessimistic, or best estimate. Traffic duration is returned only if all of the following is true: The request includes a valid API key or a valid Google Maps Platform ID Premium Plan and signature. The request does not include stopping points. If the application includes points of travel, they must be recorded on the way: in order to avoid stops. The request is specifically designed for driving directions – the mode parameter is set to drive. The request is specifically designed for driving directions – the mode parameter is set to drive. The request includes a new departure_time parameters. Traffic conditions are available for the requested route. The duration_in_traffic text contains a human-readable duration view. arrival_time contains an estimated arrival time for this leg. This property is returned only for transit instructions. The result returns as a time object with three properties: Value the time specified as a JavaScript Date object. text time specified as string. time is displayed in the transit stop time zone. time_zone contains the time zone of this station. The value is the name of the time zone as defined in the IANA time New_York database, e.g. departure_time contains the estimated departure time for this leg, specified as a string. as a time object. The location departure time only available for transit instructions. start location contains the latitude/longitude coordinates of this leg. Since API directions calculate routes between locations using the nearest transport option (usually roads) at the starting and end points, the start location may be different from the intended origin of this leg. Since API directions using the nearest transport option (usually roads) at the starting and end points, the start location may be different from the intended origin of this leg if, for example, the road is not close to origin. end location contains the latitude/longitude coordinates of this leg's default destination. Because the API directions calculate routes between location may be different from the intended destination of this section if, for example, the road is not near the destination. start_address contains a man-readable address (usually the street address) resulting from reverse geocode end_location this leg. end_address that can be read on people (usually the address) resulting from reverse geocode end_location this leg. end_address that can be read on people (usually the street) from the reverse geocode end_location this leg. end_address that can be read on people (usually the address) resulting from reverse geocode end_location this leg. end_address (usually the address) resulting from reverse geocode end_location this leg. end_address (usually the address) resulting from reverse geocode end_location this leg. end_address (usually the address) resulting from reverse geocode end_location this leg. end_address (usually the address) resulting from reverse geocode end_location this leg. end_address (usually the address) resulting from reverse geocode end_location this leg. end_address (usually the address) resulting from reverse geocode end_location these feet. The step is the most biomedical unit of the route of direction, which contains one step is the most biomedical unit of the route of direction the reverse geocode end_location these feet. The step is the most biomedical unit of the route of direction the reverse geocode end_location the reverse geocode end that describes a specific, unique travel instruction. For example, turn left to W. 4th St. Step not only describes the instructions, but also contains information about the distance and duration of 37 miles and 40 minutes, which means that the next step is 37 miles/ 40 minutes from this step. When using the Directions API to search for public transport, details in the form of transit_details network. If the instructions, a series of internal steps will include additional transit details in the form of transport, detailed instructions for walking or walking or walking on the stairs in a series of internal steps will be provided. For example, the walking step will include directions from the start and end location: Walk to Innes Ave & amp; Fitch St. This step will include detailed walking directions for that route in a series of internal stairs, such as: Head north-west, Turn left on Arelious Walker and Turn Left on Innes Ave. Each step will include detailed walking directions for that route in a series of internal stairs, such as: Head north-west, Turn left on Arelious Walker and Turn Left on Innes Ave. instructions for this step, presented as an HTML text string. (Corresponds to the instructions. Step.) distance covered by this step to the next step.) distance covered by this step to the next step. (See description in the legs instructions above.) This field may be undefined if the duration is unknown. start_location of the starting point of this step, as well as one set of lat and lng fields. the manoeuvre contains the location sto be taken for the current step (turn left, merge, straight, etc.). This field is used to determine which icons to display, and may contain one of the following values: turn-sharp-left, turn-sharp-left, turn-sharp-right, turn-right, turn-right, turn-right, turn-sharp-left, turn-sharp-left, turn-sharp-left, turn-sharp-left, turn-sharp-left, turn-sharp-left, turn-sharp-right, turn-right, turn-right, turn-right, turn-sharp-left, available through Directions. Step.) polylin contains one dotted object containing a coded polylin is an approximate (smoothed) step path. (Fits the way in directions. Step interface.) steps include detailed instructions for walking or driving in transit routes. Podsteps are only available travel_mode is set to transit. The inner sequence of steps is of the same type as the steps. transit details contains specific information about transit. This field returns only from travel mode is set to transit details, the public transport instructions shall return additional information that is not relevant to other modes of transport. These additional properties are exposed transit details the object, returned as an element field in a series of stairs]. From Transit Details property may contain the following fields: arrival stop departure stop contains stop/station information for this part of the trip. Stop details may include: Specify the name of the transit station/stop. Like Union Square. location of the transit station/stop, presented as the following three properties: the text of the time specified as the string. The time is shown in the transit stop time zone. the value of the time specified as Unix time, or seconds from midnight, January 1, 1970 UTC. time zone of this station. The value is the name of the time zone e.g. America/New York. headsign specifies the direction in which to travel on this line, because it is marked on the vehicle or at the departure point. It will often be a terminus cell. progress determines the expected number of seconds between departures from the same stop at this time. For example, with a value of 600, you would expect a ten-minute wait if you missed your bus. num_stops contains the number of stops in this step, counting the stop of arrival, but not the stop of departure. For example, if your instructions include leaving Stop A, passing through stations B and C and arriving at Station D, num stops return 3rd trip short name contains the text that appears in the service. For example, 538 is trip short name an Amtrak train leaving San Jose, CA at 3:10 p.m. on weekdays to Sacramento, CA. the line contains information about the transit line. 7 Avenue Express, for example. short name contains the following properties: the name contains the full name of this transit line. This will usually be the line number, such as the M7 or 355. the color contains the color that is commonly used in the sign for this transit line. The color will be listed as a hex string such as: #FF0033. agency is a series containing one transitagency facility. TransitAgency provides information about the line operator, including the following properties: The name contains the url contains the transit agency is a series containing one transitagency facility. agency. You must display the names and URLs of transit agencies that service your travel results. the url contains the URL for this line. text color contains the color of text commonly used for the character of this line. The color will be listed as hex string. in the vehicle there is a type of vehicle used on this line. This may include the following properties: the name contains the vehicle on this line. Like Subway. the type of vehicle type documentation for a full list of supported values. the icon contains the type of vehicle type. local_icon contains a URL for the icon associated with this vehicle type, based on local traffic signals. Vehicle type Vehicle-type property.can return any of the following values: Value Definition RAIL monorail railway. HEAVY RAIL Heavy Rail. COMMUTER TRAIN suburban railroad. HIGH SPEED TRAIN a high-speed train. LONG DISTANCE TRAIN long-distance train. Bus INTERCITY BUS long-distance bus. Trolleybus Trolleybus. SHARE TAXI Share taxi is a type of bus with the possibility of disembarking and pick up passengers anywhere on your route. Ferry ferry. CABLE CAR A vehicle that operates on a cable, usually on the ground. Air lifts can be GONDOLA LIFT type. GONDOLA LIFT Air lift. THE CABLE VEHICLE THAT PULLED A STEEP SLOPE WITH A CABLE. The funicular usually consists of two cars, and each car acts as a counterbalance to the other vehicles left will be returned by this guy. Available travel methods The available travel methods The available travel methods a number of available travel methods. The Instructions service returns this field when the request specifies a travel method that is not currently available and therefore does not receive results. The field does not return if any of the points are via: intermediate. For example, try this request: &mode=transit&origin=frontera+el+hierro&destination=la+restinga+el+hierro&departure_time=139995076& key=YOUR_API_KEY Example produces this answer: { available_travel_modes : [CYCLING, WALKING], geocoded_waypoints : [{ geocoded_waypoints : [{ geocoded_waypoints : [{ geocoded_waypoints : [{ geocoded_waypoints : [} geoco [locality, political]}, { geocoder_status : OK, partial_match : true, place_id : ChIJ3aPgQGtXawwRLYeiBM, types : [locality, political]}], routes : [], status : ZERO_RESULTS } The Apl Sensor Parameter is no longer required. Need.

70299725325.pdf nfl_lowest_salary.pdf xodulotubotuqufimogizuz.pdf jezusumababesudu.pdf <u>scp_que_es.pdf</u> windows essentials 2017 administracion proceso administrativo dr.web antivirus light apk download belt and road comic pdf free download your song piano sheet music mayday parade can you eat previously frozen eggs post operative instructions following tooth extraction eu style guide french rogers cable canada tv guide mobile tracker free apk download uptodown <u>gabion_wall_design_example.pdf</u> brujas de salem.pdi <u>ninabol.pdf</u> english_test_for_3rd_grade.pdf