



# Custom Route Extension API

## Developer's Guide

Version 1.0

# Important Information

## Notices

---

**Topics:**

This section contains document notices.

- [Legal Notices](#)
- [Document Information](#)

## Legal Notices

---

© 2017 HERE Global B.V. and its Affiliate(s). All rights reserved.

This material, including documentation and any related computer programs, is protected by copyright controlled by HERE. All rights are reserved. Copying, including reproducing, storing, adapting or translating, any or all of this material requires the prior written consent of HERE. This material also contains confidential information, which may not be disclosed to others without the prior written consent of HERE.

### Trademark Acknowledgements

HERE is trademark or registered trademark of HERE Global B.V.

Other product and company names mentioned herein may be trademarks or trade names of their respective owners.

### Disclaimer

This content is provided "as-is" and without warranties of any kind, either express or implied, including, but not limited to, the implied warranties of merchantability, fitness for a particular purpose, satisfactory quality and non-infringement. HERE does not warrant that the content is error free and HERE does not warrant or make any representations regarding the quality, correctness, accuracy, or reliability of the content. You should therefore verify any information contained in the content before acting on it.

To the furthest extent permitted by law, under no circumstances, including without limitation the negligence of HERE, shall HERE be liable for any damages, including, without limitation, direct, special, indirect, punitive, consequential, exemplary and/ or incidental damages that result from the use or application of this content, even if HERE or an authorized representative has been advised of the possibility of such damages.

## Document Information

---

### Product

Name: Custom Route Extension API

Version: Version 1.0

### Document

Name: Custom Route Extension API Developer's Guide

ID: ccd0e44-1493390492-b2dd333b

Status: FINAL

Date: 2017-Apr-28, 14:41 (GMT)

---

# Contents

---

<b>Chapter 1: Introduction</b>	<b>6</b>
What Is the Custom Route Extension API?	7
Why Use the Custom Route Extension API?	7
<b>Chapter 2: Quick Start</b>	<b>8</b>
Block a Road	9
<b>Chapter 3: Guide</b>	<b>10</b>
Key Concepts	11
Architecture	11
Overlay Description Format	11
Custom Vehicle Types	12
Acquiring Credentials	12
Constructing a Request	13
HERE Server Environments	13
Reading a Response	14
Examples	14
Submit an Overlay	14
Calculate a Route	15
Service Support	18
<b>Chapter 4: API Reference</b>	<b>19</b>
Resources	20
Upload an Overlay Map	20
Calculate Route	21
Response Status Codes	22
HTTP Status Codes	22

# Chapter 1

## Introduction

---

### Topics:

- [What Is the Custom Route E...](#)
- [Why Use the Custom Route E...](#)

This document introduces the Custom Route Extension API and:

- explains key concepts
- provides examples
- documents resources and query parameters
- documents response structures and data types

## What Is the Custom Route Extension API?

The HERE Custom Route Extension API is a REST API that allows you to request routes on a custom road network using overlay maps that you have modified.

You can submit overlay maps with modifications, such as block road links, remove restrictions from road links, or insert new road links into the network. The modifications that you submit, as part of your overlay maps, are only visible to you but not to other customers. Also, these modifications are not uploaded to the official [HERE](#) map.

## Why Use the Custom Route Extension API?

Custom Route Extension API addresses the following use cases:

**Table 1: Main features of the HERE Custom Route Extension API**

Feature	Description
Route the last mile for private roads	You can add private roads for your facilities, such as to allow your fleets to route to the destination entrance or the loading dock.
Plan for short term construction works	You can block road links that may be closed for short term construction works, and add a temporary road section, such as exits, for the duration of the construction.
Avoid certain roads	You can block certain road links that your fleet, or certain vehicle types within your fleet, should not use.
Enable roads	Customer can open certain road links for their fleet, because their vehicles have special permissions.
Consider environmental conditions	You can block roads for your fleet against certain environmental conditions, such as slippery roads from rain or melting snow, flooding or falling rocks.

## Chapter 2

# Quick Start

---

### Topics:

- [Block a Road](#)

This section illustrates how a basic request with the Custom Route Extension API works.

## Block a Road

---

Block a sequence of road links for your fleet using the following GET request. In this case, the sequence is a bridge in Frankfurt, Germany.

```
GET http://cre.cit.api.here.com/2/overlays/upload.json
?map_name=OVERLAYBLOCKROAD
&overlay_spec=[{"op":"override","shape":[[50.10765,8.68774],
[50.10914,8.68771]],"layer":"LINK_ATTRIBUTE_FCN","data":{"VEHICLE_TYPES":"0"}}]
&storage=readonly
&app_id={YOUR_APP_ID}
&app_code={YOUR_APP_CODE}
```

The response indicates whether the map overlay was submitted successfully.

```
{
  "layers": [
    "LINK_FC3",
    "LINK_ATTRIBUTE_FC3",
    "ROAD_ADMIN_FC3",
    "ROAD_GEOM_FC3"
  ],
  "response_code": "201 Created"
}
```

### Related Information

- [Constructing a Request](#)
- [Upload an Overlay Map](#)

## Chapter 3

# Guide

---

### Topics:

- [Key Concepts](#)
- [Acquiring Credentials](#)
- [Constructing a Request](#)
- [Reading a Response](#)
- [Examples](#)
- [Service Support](#)

The articles in this section provide a guide to using the Custom Route Extension API. The guide describes common use cases for the API and illustrates them with practical examples.

## Key Concepts

---

This section provides information intended to help you understand and use the Custom Route Extension API.

## Architecture

The API extends the functionality of the HERE Routing API. However, requests must be sent to a separate endpoint.

When you provide modified maps via overlays, the API compiles your changes into routable binary map format. This is the same binary map format that the Platform Data Extension API uses. The Platform Data Extension API creates layers on top of the official map layers.

The Custom Route Extension API stores your changes as overlay maps, so that you can create, modify, and delete them. Similarly, the Routing API also uses these maps.

The API stores the overlay maps in the Custom Location Extension service, where you can view your overlay maps.

**Important:** Although overlay maps are stored in the Custom Location Extension service and they are in the same format as Platform Data Extension API's binary map format, we recommend that you create, modify, and delete them using the [overlays/upload.json](#) resource only.

## Overlay Description Format

Overlay maps are described in a JSON formatted file. When you submit an overlay file, they are compiled into Platform Data Extension API layers and stored as a separate map with the layers containing the map changes.

### Contents of an Overlay Description File

An overlay description file consists of a JSON array with JSON objects. Each JSON object describes a single map modification operation.

The `op` field is a mandatory field that specifies the type of operation:

- `op : create` – Creates a new road link
- `op : override` – Changes the attribute values of an existing link attribute, link access attribute, or turn restriction attribute

The other fields of an operation are specified below.

### Overlay Create Object

You can create a new link between two existing links. Currently, you can specify the link geometry and their corresponding attributes. All other link attributes are copied from an existing neighboring link. The existing links are always cut at the point where the new link is inserted. Presently, it is not possible to insert a new link at the node of an existing link.

- `shape` – An array of shape points where each shape point is an array of two elements, latitude and longitude, in WGS-84 degree. Currently, the first and last point must each fit within 5 meters tolerance onto existing links. These points must not both end up on the same link.
- `layer` – The Platform Data Extension API layer, in which the link attributes can be changed. Currently only the `LINK_ATTRIBUTE_FCN` layer is supported.
- `data` – The map of attribute/value pairs. Currently only `VEHICLE_TYPE` is supported. For more information on Layers, Indexes, and Attributes see the [Platform Data Extension API](#) documentation.

## Overlay Override Object

You can modify attributes of an existing link, or attributes in a sequence of existing links. Currently, the following attributes are supported:

- `shape` – An array of shape points where each shape point is an array of two elements, latitude and longitude, in WGS-84 degree. The geometry must be an approximate match of the existing link's geometry; shape points can be omitted. If the geometry denotes a path of several connected links in the map, then the attributes of all these links are changed.
- `layer` – The Platform Data Extension API layer in which the link attributes need to be changed. Currently, only the `LINK_ATTRIBUTE_FCN` layer is supported.
- `data` – The map of attribute/value pairs. Currently only `VEHICLE_TYPE` is supported. For more information on Layers, Indexes, and Attributes see the [Platform Data Extension API](#) documentation.

## Custom Vehicle Types

In addition to standard vehicle types like `car` or `truck`, you can also use custom vehicle types. The steps below describe this process:

- When you are modifying an overlay, `override` operations on the `LINK_ATTRIBUTE_FCN` layer can change the vehicle types that are allowed on this particular link. You can specify standard vehicle types like `car` or `truck`, and also custom vehicle types `custom1`, `custom2`, `custom3`, or `custom4`. Use a comma separated list to add custom vehicles in addition to standard vehicles.
- For routing calls, the vehicle types `custom1`, `custom2`, `custom3` or `custom4` can be specified instead of, or in addition to (using a comma separated list), `car` or `truck`. Then the Routing API only uses the links that are allowed for (any of) the specified vehicle type(s).
- If certain vehicles in your truck fleet are allowed to use certain links, but others are forbidden, then you can override the vehicle types of these links by allowing `custom1`, additionally. For routing calls, you can either specify `mode=truck, custom1` so that the truck can use those links, or just specify `mode=truck` to get the normal behavior.

## Acquiring Credentials

---

All users of HERE APIs must obtain authentication and authorization credentials and provide them as values for the parameters `app_id` and `app_code`. The credentials are assigned per application.

This document uses the placeholder text `{YOUR_APP_ID}` and `{YOUR_APP_CODE}` as placeholders for access and authorization credentials. Replace these placeholders with your own unique application-specific credentials to access the API resources.

To obtain the credentials for an application, please visit <http://developer.here.com/plans> to register with HERE.

If you wish to explore the API, use the API Explorer at <https://developer.here.com/api-explorer>.

## Constructing a Request

A typical request to the Custom Route Extension API includes the basic elements shown in the following table and, in addition, it may contain resource-specific parameters or data.

**Table 2: Basic Request Elements**

Element	Value/Example	Description
Base URL	<code>http://cre.api.here.com</code>	Production environment
	<code>http://cre.cit.api.here.com</code>	CIT environment
		For information about the available environments, see <a href="#">HERE Server Environments</a> .
Path	<code>/2/</code>	
Resource	<code>overlays/upload.json</code>	
	<code>calculateroute.json</code>	
Application Code	<code>&amp;app_code={YOUR_APP_CODE}</code>	Substitute your own unique <code>app_code</code>
Application ID	<code>&amp;app_id={YOUR_APP_ID}</code>	Substitute your own unique <code>app_id</code>

For further information, see [Examples](#).

## HERE Server Environments

HERE provides two server environments for handling your requests: a Production environment and a Customer Integration Testing (CIT) environment.

You are required to use the CIT Environment when evaluating our products via our 90-day free trial.

To access CIT for REST APIs, amend the base URL to include an additional `cit` segment. For example, the CIT URL for this API is `http://cre.cit.api.here.com`.

HERE examples and demos use this CIT environment only to provide an illustration of how the service operates.

You are required to use the Production environment for general production use. The CIT environment must **not** be used for production.

## High Loads and Performance Testing

Neither standard server environment (CIT and Production) is designed to support high loads or performance testing. You must not conduct performance tests against the CIT or Production environments. If you need to do performance testing, [contact HERE](#) to discuss options.

## Reading a Response

The response structure of `calculateroute.json` is identical to Routing API's response.

## Examples

This section provides examples of requests reflecting typical use cases for Custom Route Extension API, along with the corresponding results.

For readability, in some responses, line breaks are used and long strings are shortened with ellipsis (...).

## Submit an Overlay

The user wants to create an overlay to modify parts of the road network, particularly close to the Alte Bruecke in Frankfurt, Germany and insert a new bridge to its West.

### Request

The request

```
GET http://cre.cit.api.here.com/2/overlays/upload.json
?map_name=OVERLAYEXAMPLE1
&overlay_spec=[{"op":"override",
"shape":[[50.10765,8.68774],[50.10914,8.68771]],
"layer":"LINK_ATTRIBUTE_FCN",
"data":{"VEHICLE_TYPES":"0"}}, {"op":"create",
"shape":[[50.10937,8.68422],[50.10807,8.68525],[50.10737,8.68387]],
"data":{"NAMES":"ENGBNDemo Road"}}]
&storage=readonly
&app_id={YOUR_APP_ID}
&app_code={YOUR_APP_CODE}
```

### Response

The response confirms a successful overlay map upload with the 201 Created response code. Within a short time, the map is available for routing.

```
{
  "layers": [
    "LINK_ATTRIBUTE_FC2",
    "LINK_FC5",
    "LINK_FC3",
    "LINK_FC2",
    "LINK_ATTRIBUTE_FC3",
    "ROAD_ADMIN_FC5",
```

```

    "ROAD_ADMIN_FC2",
    "ROAD_GEOM_FC5",
    "ROAD_ADMIN_FC3",
    "LINK_ATTRIBUTE_FC5",
    "ROAD_GEOM_FC2",
    "ROAD_GEOM_FC3"
  ],
  "response_code": "201 Created"
}

```

## Related Information

- [Constructing a Request](#) on page 13
- [Upload an Overlay Map](#) on page 20
- [Reading a Response](#) on page 14

## Calculate a Route

The user wants to compute a route from the North of the blocked bridge to the Southern side.

### Request

```

GET http://cre.cit.api.here.com/2/calculateroute.json
?waypoint0=50.10951,8.68951
&waypoint1=50.10703,8.68222
&mode=fastest;car;traffic:disabled
&overlays=MY_OVERLAY_MAP
&storage=readonly
&app_id={YOUR_APP_ID}
&app_code={YOUR_APP_CODE}

```

### Response

The response to the request contains following information blocks:

- meta information such as the timestamp and map version
- a list of waypoints, each with a `linkId` and detailed information such as its `mappedPosition`, `originalPosition`, the type of waypoint, and so on
- transport mode
- a list of maneuvers, each with a position and instruction
- a list of links, each with a shape, `remainTime`, `speedLimit`, `functionalClass`, and `_type`

```

{
  "response": {
    "metaInfo": {
      "timestamp": "2017-04-18T09:10:11Z",
      "mapVersion": "8.30.69.151",
      "moduleVersion": "7.2.201714-120544",
      "interfaceVersion": "2.6.29"
    },
    "route": [
      {
        "waypoint": [
          {
            "linkId": "+52309232",

```

```

    "mappedPosition": {
      "latitude": 50.1095094,
      "longitude": 8.6895098
    },
    "originalPosition": {
      "latitude": 50.1095099,
      "longitude": 8.68951
    },
    "type": "stopOver",
    "spot": 0.8939394,
    "sideOfStreet": "neither",
    "mappedRoadName": "Schöne Aussicht",
    "label": "Schöne Aussicht",
    "shapeIndex": 0
  },
  ... more waypoints ...

],
"mode": {
  "type": "fastest",
  "transportModes": [
    "car"
  ],
  "trafficMode": "disabled",
  "feature": []
},
"leg": [
  {
    "start": {
      "linkId": "+52309232",
      "mappedPosition": {
        "latitude": 50.1095094,
        "longitude": 8.6895098
      },
      "originalPosition": {
        "latitude": 50.1095099,
        "longitude": 8.68951
      },
      "type": "stopOver",
      "spot": 0.8939394,
      "sideOfStreet": "neither",
      "mappedRoadName": "Schöne Aussicht",
      "label": "Schöne Aussicht",
      "shapeIndex": 0
    },
    "end": {
      "linkId": "-52309625",
      "mappedPosition": {
        "latitude": 50.1070309,
        "longitude": 8.6822195
      },
      "originalPosition": {
        "latitude": 50.10703,
        "longitude": 8.68222
      },
      "type": "stopOver",
      "spot": 0.5853659,
      "sideOfStreet": "neither",
      "mappedRoadName": "Schaumainkai",
      "label": "Schaumainkai - B43",
      "shapeIndex": 37
    },
    "length": 1895,
    "travelTime": 254,
    "maneuver": [
      {

```

```

    "position": {
      "latitude": 50.1095094,
      "longitude": 8.6895098
    },
    "instruction": "Head <span class=\"heading\">west</span> on <span class=\"street\">
      Schöne Aussicht</span>. <span class=\"distance-description\">
      Go for <span class=\"length\">13 m</span>.</span>",
    "travelTime": 50,
    "length": 336,
    "id": "M1",
    "_type": "PrivateTransportManeuverType"
  },
  {
    "position": {
      "latitude": 50.1114428,
      "longitude": 8.6882436
    },
    "instruction": "Turn <span class=\"direction\">right</span> onto
description\">
      <span class=\"next-street\">Börneplatz</span>. <span class=\"distance-
      Go for <span class=\"length\">319 m</span>.</span>",
    "travelTime": 64,
    "length": 319,
    "id": "M2",
    "_type": "PrivateTransportManeuverType"
  },
  ... more maneuvers ...
],
"link": [
  {
    "linkId": "+52309232",
    "shape": [
      50.1095094,
      8.6895098,
      50.1095223,
      8.6893272
    ],
    "remainTime": 254,
    "speedLimit": 13.8888893,
    "functionalClass": 3,
    "_type": "PrivateTransportLinkType"
  },
  ... more lines ...
]
},
"summary": {
  "distance": 1895,
  "trafficTime": 350,
  "baseTime": 254,
  "flags": [
    "builtUpArea",
    "park"
  ],
  "text": "The trip takes <span class=\"length\">1.9 km</span> and <span class=\"time\">4
mins</span>.",
  "travelTime": 254,
  "_type": "RouteSummaryType"
}
}
],
"language": "en-us"
},
"warnings": [
  {

```

```
"message": "Suspicious u-turn onto link 52309195",  
"code": 2  
}  
]  
}
```

## Related Information

- [Constructing a Request](#)
- [Upload an Overlay Map](#)
- [Reading a Response](#)

## Service Support

---

If you need assistance with this or any other HERE product, select one of the following options.

- If you have a HERE representative, contact them when you have questions/issues.
- If you manage your applications and accounts through [developer.here.com](https://developer.here.com), log into your account and check the pages on the SLA report or API Health. If this does not clarify the issue, then check [stackoverflow.com/questions/tagged/here-api](https://stackoverflow.com/questions/tagged/here-api).
- If you have an evaluation plan, check [stackoverflow.com/questions/tagged/here-api](https://stackoverflow.com/questions/tagged/here-api).
- If you have questions about billing or your account, [Contact Us](#).
- If you have purchased your plan/product from a HERE reseller, contact your reseller.

# Chapter 4

## API Reference

---

### Topics:

- [Resources](#)
- [Response Status Codes](#)

This section provides descriptions of the resources, parameters, return types and error codes of the HERE Custom Route Extension API.

## Resources

This section documents the resources supported by Custom Route Extension API.

### Upload an Overlay Map

This resource uploads an overlay map that modifies route-related attributes and geometries in the map. Any subsequent routing calls take the overlay change into consideration. It may take multiple seconds until the overlay changes are effective.

Overlay maps are stored in the Custom Location Extension Service and require unique names. If an overlay map with the same name already exists, it is replaced with the one that you provide.

You can use the Custom Location Extension resources like `layers/list.json` and `layers/delete.json` to view or delete the overlay maps.

#### Resource URI

The pattern for requests to the resource is

```
GET /2/overlays/upload.json
    ?app_id={YOUR_APP_ID}
    &app_code={YOUR_APP_CODE}
    &map_name=MY_OVERLAY_MAP
    &overlay_spec=[...]
    &storage=readonly
```

For more information, see [Constructing a Request](#) on page 13.

#### Resource Parameters

Parameter	Description
map_name	mandatory string Overlay map name. Case insensitive. Must contain the string OVERLAY.
overlay_spec	mandatory string JSON structure describing the map changes. See <a href="#">Overlay Description Format</a> on page 11.
storage	mandatory string Currently only <code>readonly</code> is supported.

## Response Format

The response currently contains a list of layers the overlay map consists of. When you submit an overlay request, the response includes a list of the overlay layers in the map. However, the current response is subject to change. For information about possible changes, see the Release Notes for the future releases.

## Calculate Route

Calculates a route using map modifications from custom overlay maps.

This resource is built on top of the Routing API, but currently, it only supports a subset of the features. Hence, before replacing a Routing API `calculatoroute.json` call with this resource, make sure the features or parameters you require are supported. The parameters supported have the same meaning as those of the Routing API.

## Resource URI

```
GET /2/calculatoroute.json
  ?app_id={YOUR_APP_ID}
  &app_code={YOUR_APP_CODE}
  &overlays=MY_OVERLAY_MAP
  &mode=...
  &waypoint0=..., &waypoint1=..., ...
  &departure=...
  &storage=readonly
```

For more information, see [Constructing a Request](#) on page 13.

## Resource Parameters

Parameter	Description
overlays	optional string Overlay map name. Case insensitive. Must contain the OVERLAY string.
mode	mandatory string fastest;car;traffic:disabled OR fastest;car;traffic:disabled.
waypoint0 ... waypointN	mandatory string Comma separated waypoint latitude, longitude in WGS-84 degree. A route must consist of at least 2 waypoints (start and destination). The maximum number of waypoints is limited. However, this limit is not a fixed number but is a result of the limit for the overall processing time.
departure	optional string Format: now, yyyy-MM-dd'T'HH:mm:ss OR yyyy-MM-dd'T'HH:mm:ssX. Default: Avoid all time dependent access restrictions. If no timezone is specified, then the local timezone at the route start link is used.
storage	mandatory

Parameter	Description
	string
	Currently, only <code>readonly</code> is supported.

## Response Format

The response applies to `CalculateRouteResponseType` of the HERE Routing API with additions and modifications where overlay maps cause route changes.

For more information about `CalculateRouteResponseType`, see .

## Response Status Codes

When you submit a REST request to an endpoint, the response includes an HTTP status code. On success, the response status is `200 OK` and the response includes the data that matches the request.

On failure, the response has a different status code and may include additional information about the error. You can use this additional information to troubleshoot this issue.

For more information on the available HTTP status codes, see [HTTP Status Codes](#) on page 22

## HTTP Status Codes

In response to a request, Custom Route Extension API can return several types of status codes that indicate an error. They are shown in the table below.

Note that Custom Route Extension API supports the standard [HTTP status codes](#).

Status	Message	Error (description)
201 Created		Successful overlay map submission.
400 Bad Request		Missing Parameters or invalid parameters.
403 Forbidden		Incorrect <code>app_code</code> or <code>app_id</code> in the request. See for more information.
404 Not found		Resource not found.
500 Internal error		There is a server configuration issue.
503 Service Unavailable		Indicates that the service is temporarily unavailable due to system overload or maintenance.