



Embedded Editor

Developer's Guide

Version 1.1

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: Embedded Editor

Version: Version 1.1

Document

Name: Embedded Editor Developer's Guide

ID: 0ff0629-1496327067-19d9e61b

Status: FINAL

Date: 2017-Jun-01, 14:24 (GMT)

Contents

Chapter 1: Introduction	6
What Is the Embedded Editor?.....	7
Why Use the Embedded Editor?.....	7
Chapter 2: Quick Start	8
Add to Website.....	9
Use as a WebView Component.....	9
Chapter 3: Guide	11
Key Concepts.....	12
Acquiring Credentials.....	12
Constructing a Request.....	12
HERE Server Environments.....	13
Examples.....	13
Display Specific Use Cases.....	13
Specify Start Location for the Map.....	15
Auto-detect Mobile Device Location.....	15
Specify Language Displayed for User.....	16
Specify Zoom Level.....	17
Close the iframe.....	18
Service Support.....	18
Chapter 4: API Reference	19
Feedback.....	20

Chapter 1

Introduction

Topics:

- [What Is the Embedded Edito...](#)
- [Why Use the Embedded Edito...](#)

This document introduces the Embedded Editor and:

- explains key concepts
- provides examples
- documents resources and query parameters

What Is the Embedded Editor?

The Embedded Editor is a ready-to-use, customizable map editor that provides feedback on maps to HERE. It can be integrated into web and mobile customer environments either as an iframe within a web page or as a WebView/WKWebView component on an Android or iOS device.

Why Use the Embedded Editor?

Embedded Editor provides resources to address the following high-level use cases:

Table 1: Main features of the HERE Embedded Editor

Feature	Description
Enable feedback on maps in your web pages.	The Embedded Editor enables you to embed an iframe in your pages that allows your customers to provide feedback on HERE maps.
Enable feedback on maps in your mobile apps.	The Embedded Editor enables you to embed a WebView/WKWebView component in your app that allows your customers to provide feedback on HERE maps.
Enable feedback on maps in your mobile browser (companion application).	The Embedded Editor enables you to launch a mobile browser for providing feedback on HERE maps from your companion application.

Chapter 2

Quick Start

Topics:

- [Add to Website](#)
- [Use as a WebView Component](#)

To use the Embedded Editor, you need to embed it either as an iframe component in your web page where you host HERE maps or as a view in your application. The pages below provide information on how to integrate these components.

Add to Website

To embed the HERE Embedded Editor in a website, use an iframe. The following codeblock illustrates an example of an iframe that references the Embedded Editor.

```
<iframe
  width="600"
  height="500"
  src="https://mapfeedback.here.com?app_id={YOUR_APP_ID}&
  app_cope={YOUR_APP_CODE}&[attribute=value]">
</iframe>
```

You can use the query parameters to customize the behavior of the iframe to match your design patterns.

Related Information

- [Acquiring Credentials](#)
- [Close iframe Example](#)
- [feedback](#)
- [W3C iframe Specification](#)

Use as a WebView Component

When you use the Embedded Editor in an Android or iOS application, you can either launch the mobile device browser to load the reporting page or use the Android component `WebView` or iOS component `WKWebView`. In either case, construct the URL passed to the mobile device browser or `WebView`/`WKWebView` based on how you want the user to interact with the Embedded Editor. You need to add the relevant `app_id` and `app_code` credentials to the URL in order to use the service.

To launch a browser from an Android device, use the method `startBrowser()` in `MainActivity` as follows:

```
String feedbackUrl = "https://mapfeedback.here.com/";

// use a view intent on the URL to open the default browser
Intent browserIntent = new Intent(Intent.ACTION_VIEW, Uri.parse(feedbackUrl));
startActivity(browserIntent);
```

The following steps create an example implementation that illustrates one way to use the Android component `WebView`. If you are using an iOS application, create the functional equivalent for iOS.

1. In the application `AndroidManifest.xml` file, enable the correct permissions as follows:

```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
```

2. Configure the `WebView` to enable JavaScript and Geolocation capabilities as follows:

```
// make sure we're properly set up to do JavaScript and Geolocation
WebSettings webSettings = myWebView.getSettings();
webSettings.setJavaScriptEnabled(true);
```

```
webSettings.setGeolocationEnabled(true);
webSettings.setGeolocationDatabasePath( this.getFilesDir().getPath());
//cache the geolocation data
```

3. Configure the **WebView** to handle all URLs in the application as follows:

```
// We want the WebView to take care of all URLs and not open a browser
myWebView.setWebViewClient(new WebViewClient() {
    public boolean shouldOverrideUrlLoading(WebView view, String url) {
        return (false);
    }
});
```

The URL passed to **WebView** should be constructed to create the correct user experience.

4. Set up the **Back** button as follows:

```
// we want the back button to work properly inside the widget
@Override
public boolean onKeyDown(int keyCode, KeyEvent event) {
    if ((keyCode == KeyEvent.KEYCODE_BACK) && myWebView.canGoBack()) {

        // unless we have submitted properly, 'back' takes us back a page in the WebView
        if(!myWebView.getUrl().contains("#/submit")) {
            myWebView.goBack();
            return true;
        }
    }
    return super.onKeyDown(keyCode, event);
}
```

5. Implement logic to completely close the Activity when the URL contains a "#/submit" as this is when the user has submitted a feedback request.

Related Information

- [Acquiring Credentials](#)
- [Examples](#)
- [feedback](#)

Chapter 3

Guide

Topics:

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

The articles in this section provide a guide to using the Embedded Editor. 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 Embedded Editor.

Submitting Feedback

Any content that you submit via the Embedded Editor is reviewed by HERE before the data in the HERE maps is updated.

Map Versions

HERE regularly updates its maps. While the Embedded Editor uses the most current map available, users may still encounter older maps where updates have not been made.

Customization

You can customize the Embedded Editor using the *parameters* to control the user experience in terms of UI, use cases, and languages.

Acquiring Credentials

All users of the Embedded Editor must obtain authentication and authorization credentials. The authentication must be provided as values for the parameters `app_id` and `app_code`. The credentials are assigned per application.

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

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

Constructing a Request

A request to the Embedded Editor includes the basic elements shown in the following table and may additionally contain parameters to customize the UI behavior and appearance.

Table 2: Basic request elements

Element	Value/Example	Description
Base URL	<code>https://mapfeedback.here.com</code>	Production environment only
	<code>https://stg.mapfeedback.here.com</code>	CIT environment.
		For information about the available environments, see HERE Server Environments .

Element	Value/Example	Description
Resource	feedback	
Application Id	app_id	Substitute your own unique app_id
Application Code	app_code	Substitute your own unique app_code

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.

To access CIT for the Embedded Editor use the following URL: <https://stg.mapfeedback.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.

Examples

This section provides examples of requests along with the responding results.

Display Specific Use Cases

The user wants to see the feedback options only for the relevant cases.

Request

For both iframe and application solutions, the `features` parameter defines which feedback cases the Embedded Editor displays. Pass the following URL into your solution.

```
https://stg.mapfeedback.here.com?  
app_id={YOUR_APP_ID}&  
app_code={YOUR_APP_CODE}&  
features=road,place,address
```

If you want to only offer an opportunity to provide feedback on a place, use the following feature key/value pair: `features=place`

Note: In order to request all available use cases, you need to leave out the `features` parameter.

Response

Figure 1: Multiple Feedback Options

Report a map problem

Kind of problem?

Road problem

Address problem

[Place problem](#)

Figure 2: One Feedback Options

1 — 2 — 3 Place problem

What's wrong?

Place is missing

Place info is wrong

Place is closed

Report as inappropriate

Related Information

- [Constructing a Request](#)
- [feedback](#)

Specify Start Location for the Map

The user wants to see the map of the Embedded Editor at a location that is relevant to them.

Request

For both iframe and application solutions, the `coord` and `zoomLevel` parameters combine to define the context in which the Embedded Editor opens. Pass the following URL into your solution.

```
https://stg.mapfeedback.here.com?  
app_id={YOUR_APP_ID}&  
app_code={YOUR_APP_CODE}&  
coord=52.531137730274494,13.385125160713187&zoomLevel=19
```

The location is a segment of Invalidenstrasse in Berlin, Germany.

Response

Figure 3: Open

Report a map problem

Kind of problem?

Road problem

[Address problem](#)

Place problem

Related Information

- [Constructing a Request](#)
- [feedback](#)

Auto-detect Mobile Device Location

The user wants to provide feedback on a place and to have the Embedded Editor launch in a manner that is aware of the location of a mobile device.

Request

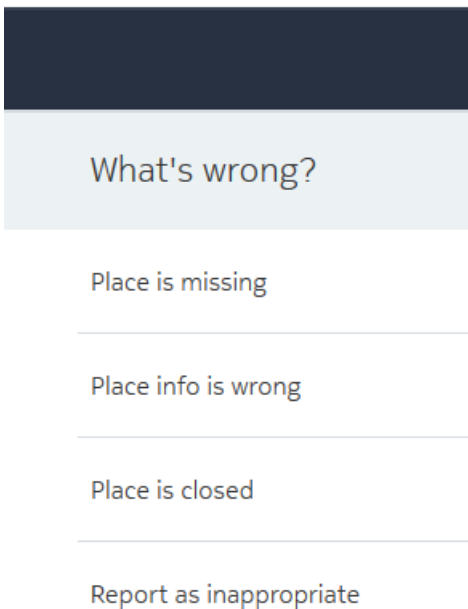
If you do not specify a `coord` parameter when on a mobile device, the Embedded Editor uses the automatically detected geolocation. This requires that the `WebView/WKWebView` has been configured to have access to the geolocation data. Pass the following URL into your solution.

```
https://stg.mapfeedback.here.com?
  app_id={YOUR_APP_ID}&
  app_code={YOUR_APP_CODE}&
  features=place
```

Response

Figure 4: Feedback for a Missing Place

1 2 3 Place problem



Related Information

- [Constructing a Request](#)
- [feedback](#)
- [Use as a WebView Component](#)

Specify Language Displayed for User

The user wants to use the same language you are in using in the larger context.

Request

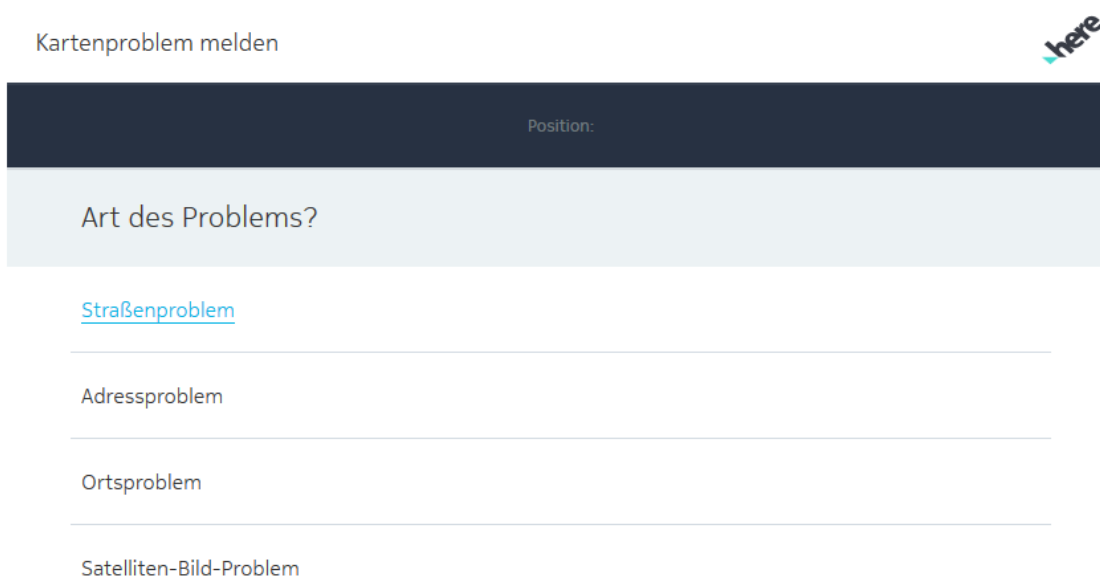
For both iframe and application solutions, the `lang` parameter defines which language the Embedded Editor displays. Pass the following URL into your solution.

```
https://stg.mapfeedback.here.com?
  app_id={YOUR_APP_ID}&
  app_code={YOUR_APP_CODE}&
  lang=de-DE
```

If you do not specify a language, the Embedded Editor uses English

Response

Figure 5: Embedded Editor with German text



Related Information

- [Constructing a Request](#)
- [feedback](#)

Specify Zoom Level

The user wants to see the map view in the Embedded Editor at a specific zoom level.

Request

For both iframe and application solutions, the `zoomLevel` parameter defines the zoom level shown by the Embedded Editor. Pass the following URL into your solution.

```
https://stg.mapfeedback.here.com?
  app_id={YOUR_APP_ID}&
  app_code={YOUR_APP_CODE}&
  zoomLevel=19
```

The larger the number (max 21), the closer the map is zoomed-in to the desired location. You can select objects on the map starting at `zoomLevel=16`. This is the minimum number if you want to work with preselected objects.

Related Information

- [Constructing a Request](#)
- [feedback](#)

Close the iframe

The user wants the iframe to close after submitting feedback.

The code below shows how to close the iframe after a successful feedback submission by clicking the x in the UI. You must use the `showClose` parameter in the request URL for the code below to work.

```
var closeIFrame= function(event)
{
  if (!event || event.data != "FeedbackWidgetClose")
    return;
  fifToggle();
}

window.addEventListener("message", closeIFrame, false);
```

Related Information

- [Add to Website](#)
- [Constructing a Request](#)
- [Feedback](#)

Service Support

If you need assistance with this or other HERE products, contact your HERE representative or Technical Customer Support.

Chapter 4

API Reference

Topics:

- [Feedback](#)

This section provides descriptions of the resources and parameters of the HERE Embedded Editor.

Feedback

The resource feedback allows you to specify how the Embedded Editor behaves when opened in the context from which it is launched. The only mandatory parameters are `app_id` and `app_code`.

Table 3: Resource Parameters

Parameter	Type	Description
<code>app_id</code>	String	A 20 byte Base64 URL-safe encoded string used for the authentication . You must include an <code>app_id</code> and <code>app_code</code> with every request. For more information, see Acquiring Credentials on page 12.
<code>app_code</code>	String	A 20 byte Base64 URL-safe encoded string used for the authentication . You must include an <code>app_id</code> and <code>app_code</code> with every request. For more information, see Acquiring Credentials on page 12.
<code>coord</code>	String	Comma separated latitude and longitude pair in WGS-84 format of the map position that sets the context for Embedded Editor. The nearest map object to this coordinate is preselected when the user first sees a map. Example: "coord=46.456,8.2356" (latitude,longitude) If no <code>coord</code> parameter is set, the device location is used and the browser asks you for permission.
<code>features</code>	String	Specifies the feedback types shown by the Embedded Editor. If you only specify only one value, the entry page starts directly at the specified feature. The supported values are as follows: <ul style="list-style-type: none"> • address • border • other • place • road • satellite Example: <code>&features=road,address,place</code> The example above enables feedback for roads, house numbers, and places. This makes it possible to send feedback about existing elements (modify, remove) or new elements.
<code>hideLogo</code>	-	By default the HERE logo is shown on the top right corner of the Embedded Editor page. Specify the <code>hideLogo</code> parameter to hide this logo.
<code>lang</code>	String	Defines the language offered by the Embedded Editor. The supported languages are: <ul style="list-style-type: none"> • "en" : English (Great Britain), default • "cs-CZ": Czech (Czech Republic) • "de-DE": German (Germany) • "el-GR": Greek (Greece) • "es-ES": Spanish (Spain) • "fr-FR": French (France) • "hu-HU": Hungarian (Hungary)

Parameter	Type	Description
		<ul style="list-style-type: none"> "it-IT": Italian (Italy) "nl-NL": Dutch (Netherlands) "ms-MY": Malay (Malaysia) "pl-pl": Polish (Poland) "pt-BR": Portuguese (Brazil) "ro-RO": Romanian (Romania) "ru-RU": Russian (Russia) "sk-SK": Slovak (Slovakia) sv-SE": Swedish (Sweden) "th-TH": Thai (Thailand) "tr-TR": Turkish (Turkey) "zh-CN": Chinese (China)
showClose	-	Specify the showClose parameter to display a close button (x) on the top right corner of the application. When users click the close button, a message is sent to the parent window via postMessage ("FeedbackWidgetClose", '* '); This enables the parent window to close the tag or iframe of the MapFeedback widget.
webView	String	<p>If the Embedded Editor is running within a WebView, this parameter defines the platform of the WebView.</p> <p>Valid WebView platforms:</p> <ul style="list-style-type: none"> Android - Example: ?webView=Android For more information, see https://developer.android.com/reference/android/webkit/WebView.html WKWebView (iOS) - Example: ?webView=WKWebView For more information, see https://developer.apple.com/reference/webkit/wkwebview
zoomLevel	Integer	Defines the zoom level at which the Embedded Editor should start. "0" is zoomed out most and "21" is zoomed in the most. The map uses image tiles from "0" to "16" and vector data at "17" and higher.