

I

Zero Code Backend API

Design Document



© Copyright 2020 Manthan Software Services Pvt. Ltd. All rights reserved.

Manthan™ is a trademark of Manthan Software Services Pvt. Ltd.

No portion of this document may be reproduced in whole or in part without the prior written permission of Manthan. All written materials are protected by copyright laws.

Manthan makes every effort to keep the information contained in this document current and accurate, however Manthan makes no warranty or representation, express or implied, that the information contained in this document is accurate or complete. No person should consider our distribution of this material as making any representation or warranty with respect to such material and should not rely upon it as such.

This document is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract either explicitly or by implication. It is not a commitment to deliver any material, code, or functionality, and must not form the basis for purchase decisions. The development, release, and timing of any features or functionality, whether or not described herein, for Manthan products remains at the sole discretion of Manthan.

Table of Contents

1. ABOUT THIS DOCUMENT	1
1.1. SCOPE.....	1
1.2. AUDIENCE.....	1
2. HTTP RESPONSE CODES.....	2
3. ZERO CODE BACKEND APIS	3
3.1. FETCH ALL ACTIVE JIWS.....	3
3.2. ADD JIW INFORMATION	5
3.3. DELETE JIW INFORMATION	6
3.4. PUBLISH JIW TO PRODUCTION/QA	7
3.5. PUBLISH JIWS TO BUILD SERVER.....	8
3.6. VIEW JIW VERSION.....	9
3.7. SAVE JIW VERSION	10
3.8. DISCARD JIW VERSION	10
3.9. CHANGE LOG VIEW	12
3.10. ROLLBACK JIW VERSION.....	14
3.10. FETCH ALL DOMAINS.....	15
4. SQL TABLES	16
5. VERSION CONTROL.....	18

1. About This Document

This document provides information on the APIs available for Zero Code.

1.1. Scope

Scope of this document includes the request and response details of each API, and HTTP response codes available for Zero Code.

1.2. Audience

The intended audience consists of programmers involved backend API and dashboard developers.

2. HTTP Response Codes

Response Code	Description
200 OK	The request is successful.
400 BAD REQUEST	Invalid parameter or missing required parameter in the request.
405 Method not allowed	Supports only HTTP specific method.
404 NOT FOUND	Page not found.

3. Zero Code Backend APIs

The following APIs are used for Zero Code:

- Fetch All Active JIWs
- Add JIW Information
- Delete JIW Information
- Publish JIW to Production/QA
- Publish JIWs to Build Server
- View JIW Version
- Save JIW Version
- Discard JIW version
- Rollback JIW Version
- Fetch All Domains

3.1. Fetch All Active JIWs

Fetch All Active JIWs API is used to fetch all active JIWs for the selected site on the dashboard.

How to Invoke

URI	zerocode/fetchAllJiw
HTTP Method	GET

Returns the array of all active JIWs with selected **site_id**.

Response Parameters

Parameter	Description
jiwId	Auto increment ID of the JIW.
jiwName	JIW name.
page_url	JIW indicative page URL.
prodVersion	Current production deployed version.
qaVersion	Current QA deployed version.
stage	Current stage of the JIW (dev, qa, or prod).
pageTypeid	Page Type of the JIW.

Sample Response

```
{
  "jiws":[
    {
      "jiwId":1,
      "name":"Home Page",
      "pageUrl":"https://example.com",
      "prodVersion":"1.0",
      "qaVersion":"0.7",
      "Stage":"prod"
    },
    {
      "jiwId":2,
      "name":"Category Page",
      "pageUrl":"https://example.com/mens/shirt.html",
      "prodVersion":"0.0",
      "qaVersion":"0.2",
      "stage":"dev"
    },
    {
      "jiwId":3,
      "name":"Item Page",
      "pageUrl":"https://example.com/shirt/mys-shirt.html",
      "prodVersion":"2.0",
      "qaVersion":"1.3",
      "stage":"qa"
    }
  ]
}
```

3.2. Add JIW Information

Add JIW Information API is used to add new JIW with the required details such as name, description, and the indicative URL to refer.

How to Invoke

URI	zerocode/addJiw
HTTP Method	POST

Request Parameters

Parameter	Description
name	JIW name specified on the UI.
description	Description of the JIW.
pageUrl	Indicative page URL to refer.
pageTypeId	Selected Page Type

Sample Request

```
{
  "name": "Cart Page",
  "description": "Cart Page",
  "pageUrl": "https://example.com/cart",
  "pageTypeId": 13
}
```

Sample Response

```
{
  success: {
    message code & message
  },
  error: {
    message if any
  }
}
```



```
}  
}
```

3.3. Delete JIW Information

Delete JIW Information is used to maintain the soft delete at the row level with an active status flag in the table.

How to Invoke

URI	zerocode/deleteJiws
HTTP Method	POST

Sample Request

```
{  
  "jiwIds": [  
    1,  
    3,  
    5  
  ]  
}
```

Sample Response

```
{  
  success: {  
    message code & message  
  },  
  error: {  
    message if any  
  }  
}
```

3.4. Publish JIW to Production/QA

Publish JIW to Production/QA API is used to publish one or more JIWs to different stages. Stages can be moved in an order of dev to QA and QA to production. All the drafted JIWs can moved to QA and then from QA to production. It will create a new version entry increment by 0.1 in QA stage and increment by 1 in the production stage. Once it is saved, then it will be called in the **Publish JIWs to Build Server** API to publish the JIW.

How to Invoke

URI	zerocode/publish
HTTP Method	POST

Sample Request

```
{
  "jiwIds":[
    1,
    3,
    5
  ],"stage":"1/2"
}
```

Sample Response

```
{
  success:{
    message code & message
  },
  error:{
    message if any
  }
}
```

```
}
}
```

3.5. Publish JIWs to Build Server

Publish JIWs to Build Server API is used to publish all the production or QA JIWs to the build server.

URI	build_server_path/build
HTTP Method	POST

Request Parameters

Parameter	Description
siteld	Site ID to publish the JIWs.
stage	Stages will be QA or Production.
jiwConfig	Object consists of both page load and Ajax interceptor JIWs.
onPageLoadFns	Array of all the page load JIW content for the selected site. Based on the stage either QA or Production.
ajaxInterceptorFns	Array of all the Ajax interceptor JIW content for the select site. Based on the stage either QA or Production.
siteConfig	Object contains all the site related configurations such as API Key, Clint Key, and zero code enable flag.

Sample Request

```
{
  "siteId": "123",
  "jiwConfig": {
    "ajaxInterceptorFns": [],
    "onPageLoadFns": []
  },
  "siteConfig": {
    // Add all the site config that might be needed by the specific site to be used in
    low code build
  }
}
```

Sample Response

```
{
```

```

    success:{
        message code & message
    },
    error:{
        message if any
    }
}

```

3.6. View JIW Version

View JIW Version API is used to view and edit the current version of JIW to update the information of both page load and Ajax interceptor JIW content. It requires the **jiwId** as a request parameter.

URI	zerocode/viewJiwVersion/{jiwId}
HTTP Method	GET

Sample Response

```

{
    "jiwId" : "123",
    "pageLoadJiw":"Test Script content on page load JIW",
    "ajaxInterceptorJiw" : "Ajax Interceptor Content."
    "stage":"dev/qa/prod"
}

```

3.7. Save JIW Version

Save JIW Version API is used to save the JIW version content of the dev stage.

URI	zerocode/saveJiwVersion
HTTP Method	POST

Sample Request

```
{
  "jiwId" : "123",
  "pageLoadJiw": "Test Script content on page load JIW",
  "ajaxInterceptorJiw" : ""
}
```

Sample Response

```
{
  success:{
    message code & message
  },
  error:{
    message if any
  }
}
```

3.8. Discard JIW Version

Discard JIW Version API is used to discard the current dev version of the JIW content of given **jiwId**.

URI	zerocode/discardJiw
HTTP Method	POST

Sample Response

```
{  
  success:{  
    message code & message  
  },  
  error:{  
    message if any  
  }  
}
```

3.9. Change Log View

Change Log View API is used to get the all the Prod and QA version of the JIW.

URI	zerocode/fetchJIWVersionLogs/{JiwId}
HTTP Method	GET

Sample Response

```
{
  "data": [
    {
      "id": 10,
      "parentId": 0,
      "versionCode": "2.0",
      "date": "12/28/2020 15:11",
      "pageload_jiw": "My Test Content",
      "ajax_interceptor_jiw": "Ajax Content 1"
    },
    {
      "id": 9,
      "parentId": 10,
      "versionCode": "1.3",
      "date": "12/28/2020 15:11",
      "pageload_jiw": "My Test Content",
      "ajax_interceptor_jiw": "Ajax Content 1"
    },
    {
      "id": 8,
      "parentId": 10,
      "versionCode": "1.2",
      "date": "12/28/2020 15:11",
      "pageload_jiw": "My Test Content",
      "ajax_interceptor_jiw": "Ajax Content 1"
    }
  ],
}
```

```
{
  "id":7,
  "parentId":10,
  "versionCode":"1.1",
  "date":"12/28/2020 15:11",
  "pageload_jiw":"My Test Content",
  "ajax_interceptor_jiw":"Ajax Content 1"
},
{
  "id":6,
  "parentId":0,
  "versionCode":"1.0",
  "date":"12/28/2020 15:11",
  "pageload_jiw":"My Test Content",
  "ajax_interceptor_jiw":"Ajax Content 1"
},
{
  "id":5,
  "parentId":6,
  "versionCode":"0.5",
  "date":"12/28/2020 15:11",
  "pageload_jiw":"My Test Content",
  "ajax_interceptor_jiw":"Ajax Content 1"
},
{
  "id":4,
  "parentId":6,
  "versionCode":"0.4",
  "date":"12/28/2020 15:11",
  "pageload_jiw":"My Test Content",
  "ajax_interceptor_jiw":"Ajax Content 1"
},
}
```



```
{
  "id":3,
  "parentId":6,
  "versionCode":"0.3",
  "date":"12/28/2020 15:11",
  "pageload_jiw":"My Test Content",
  "ajax_interceptor_jiw":"Ajax Content 1"
},
{
  "id":2,
  "parentId":6,
  "versionCode":"0.2",
  "date":"12/28/2020 15:11",
  "pageload_jiw":"My Test Content",
  "ajax_interceptor_jiw":"Ajax Content 1"
},
{
  "id":1,
  "parentId":6,
  "versionCode":"0.1",
  "date":"12/28/2020 15:11",
  "pageload_jiw":"My Test Content",
  "ajax_interceptor_jiw":"Ajax Content 1"
}
]
```

3.10. Rollback JIW Version

Rollback JIW Version API is used to revert JIW versions of QA or Production version to the current dev.

URI	zerocode/rollbackjiw
HTTP Method	POST

Sample Request

```
{
  "jiwId" : "123",
  "version":"2.0",
  "stage" : "qa/prod"
}
```

Sample Response

```
{
  success:{
    message code & message
  },
  error:{
    message if any
  }
}
```

3.10. Fetch All Domains

Fetch All Domains API is used to return array of domains for the selected Site ID.

URI	zerocode/fetchDomains
HTTP Method	POST

Sample Response

```
{
  ["domain1.com","domain2.com"]
}
```

4. SQL Tables

You need two tables to maintain the JIW data. First table is a master table for all the JIW with the indicative reference URLs and the second table is for details to all the versions of each JIW with different stages.

PAGE_JIWS		
Java Script Integration Widget for Pages		
Column	Type	Description
jiw_id	Serial	Specifies the auto increment primary key column.
jiw_name	character varying (50)	Specifies the JIW name.
description	Text	Specifies the description of the JIW.
page_url	Text	Specifies the indicative page URL to write the JIW code.
active	byte	Specifies the byte value whether the page is active. Default is 1. Once deleted, it will be 0.
site_id	Integer	Specifies the selected site id.
modified_date	Datetime	Specifies the last modified timestamp.

PAGE_JIW_VERSIONS		
JavaScript Integration Widget of all Versions with Different Stages		
Column	Type	Description
jiw_version_id	BigSerial	Specifies the auto increment primary key column.
jiw_id	integer	Specifies the reference ID from the PAGE_JIW table.
pageload_jiw	Text	Specifies the page load JIW.
ajax_interceptor_jiw	Text	Specifies the Ajax interceptor JIW.
jiw_stage	smallint	Specifies the different type of stages. <ul style="list-style-type: none"> dev -0 qa - 1 prod - 2 Default will be 0.
version_id	Integer	Specifies running number for the version of the JIW.

PAGE_JIW_VERSONS		
version_code	numeric	Specifies the version numbers for both QA and Prod stages. QA versions will be incrementing by minor version 0.1 and prod versions incrementing with 1.
latest	byte	Specifies the latest content of JIW. Value 1 will be the current active for the stage.
modified_date	datetime	Specifies the last modified timestamp.
activated_date	datetime	Specifies the date of publish to QA or Production.

DOMAIN_NAMES		
List of Whitelisted Domains for the Site		
Column	Type	Description
site_id	integer	Specifies the selected site id.
Domain_url	text	Specifies the domain URL to be whitelisted for the site ID.

Version	Date	Author	Reviewer	Comments
1.0	27-Nov-2020	Senthilkumar R		First draft

5. Version Control