File Exchange API Documentation

# POST - /pcb/api/uploadAndExtract

API to upload ZIP file for design checks and storing for future use.

## Request Parameters

1 – File **(required)**

2– ProjectId (i.e FolderName) **(required)**

3 – CustomerId

4 – ServiceType

File is the important field, in which user needs to choose a zip file which he/she wants to extract and Project Id is the Folder Name In which your files will extract. After calling the Upload and Extract API the FileExtractUploadService is called which does all the processing on the file that the user will upload and categorize all the files according to the file format like Gerber files, Drill files, ODB formats and many more.

**Request Syntax**

{

projectId: "String",

customerId: "String",

serviceType: "String",

uploadFile: File

}

## Response Syntax

[

{

"key": {

"projectId": "String",

"version": Integer

},

"customerId": String,

"serviceTypeId": Integer,

"summary": String",

"fileTypeToFileNameMapping": {

"1": "String",

"2": "String",

"3": "String"

},

"errors": [

"String Array"

],

"notes": [

"String Array"

],

"projectFiles": [

"String Array”

],

"status": "String",

"version": Integer,

"projectId": "String"

}

]

# /pcb/admin/extensions

This API is a GET Mapping API which is very simple to use just hit the above URL and the result will be displayed to the user in the JSON Response. The result contains all the extensions that are contained in the file with its id that user have uploaded while creating extensions. This API will call the FileExtensionService which has findAll() method for getting all the file extensions.

**Response Syntax**

[

{

"id": Integer,

"name": "String"

}

]

# POST - /pcb/admin/extensions/create

This API is a POST Mapping API which takes the JSON Response. The JSON contains id and extension in the request.

The id contains Integer values while name contains String values.

This API will call the FileExtensionService which has save () method for saving the file extensions. Status: 200 OK which means our API is working properly:-

## Request Syntax

{

"id": Integer,

"name": "Integer"

}

# GET - /pcb/admin/extensions/retrive/{id}

This API is a GET Mapping API which takes argument in the URL which is the extensionId that you want to get from the database. The output will be In JSON format which contains id and extensions in the response. This API will call the FileExtensionService which has getExtensionById () method for retrieving the file extensions.

## Response Syntax

{

"id": Integer,

"name": "String"

}

# GET - /pcb/admin/services/

This API is a GET Mapping API which is a very simple API; User just needs to hit the URL and the output will be In JSON format which contains id and name of the service in the response.

The Id contains Integer values while name contains String values.

The Get Services API will call the FileServices which has a findAll() method which gets all the id and services name. The output is shown below:-

## Response Syntax

[

{

"id": Integer,

"name": "String"

}

]

# POST - /pcb/admin/services/create

This API is a POST Mapping API which takes the JSON Response. The JSON contains id and service name in the request.

The Id contains Integer values while name contains String values.

This API will call the FileService which has save () method for saving the file services. Below shows the Request Syntax of the API that is Status: 200 OK which means our API is working properly:-

## Request Syntax

{

"id": Integer,

"name": "String"

}

# GET - pcb/admin/services/retrive/{id}

This API is a GET Mapping API which takes argument in the URL I.e. service Id of the service which you want as a response.

The id contains Integer values and name contains String values.

This API calls the FileService which has getServicesById() method for retrieving the FileServices.

## Response Syntax

{

"id": Integer,

"name": "String"

}

1. GET - /pcb/admin/filetypes/

This API is a GET Mapping API which is a very simple API; User just needs to hit the URL and the output will be In JSON format which contains id and name of file extensions in the response.

The id contains Integer values and type contains String values.

This API calls the FileTypeService which has findAll() method to get all the filetypes. Below shows the Response returned by the API in JSON format.

## Response Syntax

[

{

"id": Integer,

"type": "String"

}

]

# POST - /pcb/admin/filetypes/create

This API is a POST Mapping API which takes the JSON Response. The JSON contains id and filetype in the request.

The id contains Integer values while type contains String values.

This API will call the FileTypeService which has save () method for saving the file type. The output of the API will be Status: 200 OK which means our API is working properly:-

## Request Syntax

{

"id": Integer,

"type": "String"

}

# GET - /pcb/admin/filetypes/retrive/{id}

This API is a GET Mapping API which takes argument in the URL I.e. filetype Id of the filetype which you want as a response.

The id contains Integer values and type contains String values.

This API calls the FileTypeService which has getTypeById() method for retrieving the Filetypes. Below image shows the output of the API:-

## Response Syntax

{

"id": Integer,

"type": "String"

}

# GET - /pcb/admin/servicefiles

This API is a GET Mapping API which is a very simple API; User just needs to hit the URL and the output will be In JSON format which contains composite key of serviceId and filetypeId , service, file, filetypeId, serviceId of servicefiles in the response.

The key contains Integer values while service and file contains String values. Again serviceId and filetypeId contains integer values.

This API calls the ServiceFilesService which has findAll() method to get all the servicefiles.Below image shows the output in JSON format:-

## Response Syntax

[

{

"key": {

"serviceId": Integer,

"filetypeId": Integer

},

"service": "String",

"file": "String",

"serviceId": Integer,

"filetypeId": Integer

}

]

# POST - /pcb/admin/servicefiles/create

This API is a POST Mapping API which takes the JSON Response. The JSON contains composite key of serviceId and filetypeId , service, file, filetypeId, serviceId in the request.

The Key contains Integer values while service and file contains String values. serviceId and filetypeId again both takes Integer values.

This API will call the ServiceFilesServices which has save () method for saving the Service Files. The output of the API that is Status: 200 OK which means our API is working properly:-

## Request Syntax

{

"key": {

"serviceId": Integer,

"filetypeId": Integer

},

"service": "String",

"file": "String",

"serviceId": Integer,

"filetypeId": Integer

}

# GET - /pcb/admin/servicefiles/retrive/{ serviceId }

This API is a GET Mapping API which takes argument in the URL I.e. service Id of the ServiceFiles which you want as a response.

The Key contains Integer values while service and file contains String values. ServiceId and filetypeId again both contains Integer values.

This API calls the ServiceFilesServices which has getFilesByService() method for retrieving the ServiceFiles. Below shows the Request and Response of the API:-

## Response Syntax

[

{

"key": {

"serviceId": Integer,

"filetypeId": Integer

},

"service": "String",

"file": "String",

"serviceId": Integer,

"filetypeId": Integer

}

]

# POST - /pcb/admin/servicefiles/createmulti

This API is a POST Mapping API which takes the Array of JSON Response. The JSON contains a composite key of serviceId and filetypeId , service, file,filetypeId, serviceId in the request.

The Key contains Integer values while service and file contains String values. ServiceId and filetypeId again both contains Integer values.

This API will call the ServiceFilesServices which has saveAll () method for saving the Service Files. Below shows the Request of the API .The Response would be Status: 200 OK which means our API is working properly:-

## Request Syntax

[

{

"key": {

"serviceId": Integer,

"filetypeId": Integer

},

"service": "String",

"file": "String",

"serviceId": Integer,

"filetypeId": Integer

},

{

"key": {

"serviceId": Integer,

"filetypeId": Integer

},

"service": "String",

"file": "String",

"serviceId": Integer,

"filetypeId": Integer

}

]

# GET - /pcb/admin/extensionfiles

This API is a GET Mapping API which is a very simple API; User just needs to hit the URL and the output will be In JSON format which contains a composite key of extensionId and filetypeId, extension, file, filetypeId, extensionId of extensionFiles in the response.

The Key contains Integer values while file and extension contains String values. extensionId and filetypeId again both contains Integer values.

This API calls the ExtensionFileService which has findAll() method to get all the extensionfiles.

## Response Syntax

[

{

"key": {

"extensionId": Integer,

"filetypeId": Integer

},

"extension": "String",

"file": "String",

"filetypeId": Integer,

"extensionId": Integer

}

]

# POST - /pcb/admin/extensionfiles/create

The API is a POST Mapping API which takes the JSON Response. The JSON contains composite key of extensionId and filetypeId , extension, file, filetypeId, extensionId in the request.

The Key contains Integer values while file and extension contains String values. extensionId and filetypeId again both contains Integer values.

This API will call the ExtensionFileService which has save () method for saving the Extension Files.

## Request Syntax

{

"key": {

"extensionId": Integer,

"filetypeId": Integer

},

"extension": "String",

"file": "String",

"filetypeId": Integer,

"extensionId": Integer

}

# POST - /pcb/admin/extensionfiles/createmulti

This API is a POST Mapping API which takes the Array JSON Response. The JSON contains composite key of extensionId and filetypeId , extension, file, filetypeId, extensionId in the request.

The Key contains Integer values while file and extension contains String values. extensionId and filetypeId again both contains Integer values.

This API will call the ExtensionFileService which has saveAll () method for saving the Extension Files. API return a Response Status: 200 OK which means our API is working properly:-

## Response Syntax

[

{

"key": {

"extensionId": Integer,

"filetypeId": Integer

},

"extension": "String",

"file": "String",

"filetypeId": Integer,

"extensionId": Integer

},

{

"key": {

"extensionId": Integer,

"filetypeId": Integer

},

"extension": "String",

"file": "String",

"filetypeId": Integer,

"extensionId": Integer

}

]

# GET - /pcb/admin/extensionfiles/retrive/{fileTypeId }

This API is a GET Mapping API which takes argument in the URL I.e. filetype Id of theExtensionFiles which you want as a response.

The Key contains Integer values while file and extension contains String values. extensionId and filetypeId again both contains Integer values.

This API calls the ExtensionFileService which has getExtenFileTypeById () method for retrieving the ExtensionFiles.

## Response Syntax

[

{

"key": {

"serviceId": Integer,

"filetypeId": Integer

},

"service": "String",

"file": "String",

"serviceId": Integer,

"filetypeId": Integer

},{

"key": {

"serviceId": Integer,

"filetypeId": Integer

},

"service": "String",

"file": "String",

"serviceId": Integer,

"filetypeId": Integer

}

]

# GET - /pcb/admin/report/

This API is a GET Mapping API which returns the whole report; User just needs to hit the URL and the output will be In JSON format which contains a composite key of projectId and version, customerId, serviceTypeId, summary, fileTypeToFileNameMapping, errors, notes, projectFiles, status, version, projectId in the response.

Every key takes some type of parameter which is shown in the response below.

This API calls the ReportServices which has findAll() method to get all the reports.

## Response Syntax

[

{

"key": {

"projectId": "String",

"version": Integer

},

"customerId": String,

"serviceTypeId": Integer,

"summary": String",

"fileTypeToFileNameMapping": {

"1": "String",

"2": "String",

"3": "String"

},

"errors": [

"String Array"

],

"notes": [

"String Array"

],

"projectFiles": [

"String Array”

],

"status": "String",

"version": Integer,

"projectId": "String"

}

]