Integration

In this section, a detailed explanation for integrating Synap Editor into your system will be provided.
Try integration with a variety of development environments using reference guides.

- Uploading an image file
  - [request]
  - [response]
- Uploading a video clip
  - [request]
  - [response]
- Uploading an arbitrary file
  - [request]
  - [response]
- Import Document
  - [request]
  - [response]

Sample Implementations

- Java Spring Framework Example
- Java Servlet Example
- ASP.NET (C#) Example
- ASP(Classic) Example
- PHP Example
- PHP4 Example
- Django Example
- Ruby On Rails Example
- Wordpress plugin Example

All examples assume that the editor instance was configured as follows:

Synap Editor Configuration

```
new SynapEditor('synapEditor', {
        'editor.upload.image.api': "/uploadImage",
        'editor.upload.video.api': "/uploadVideo",
        'editor.upload.file.api': "/uploadFile",
        'editor.import.api': "/importDoc"
});
```

Uploading an image file

[request]

Send image file to your system via the URL that was configured as '`editor.upload.image.api`' property.

```
Request URL: http://hostname/uploadImage
Request Method: POST
```

Key	Description	Required	Type
file	image file to be uploaded (multipart/form-data)	O	Binary

[response]

Once the image has been successfully uploaded, your system should respond with the following data:

```
{
    "uploadPath":"uploads/36a43f36f442b5824c6b061eb734553d.png"
}
```

Key	Description	Required

uploadPath	path to the uploaded image (as a URL that can be accessed from a web browser)	String

Uploading a video clip

[request]

Send video file to your system via the URL that was configured as `'editor.upload.video.api'` property.

```
Request URL: http://hostname/uploadVideo
Request Method: POST
```

Key	Description	Required	Type
file	video file to be uploaded (multipart/form-data)	O	Binary

[response]

Once the video has been successfully uploaded, your system should respond with the following data:

```
{
    "uploadPath":"uploads/36a43f36f442b5824c6b061eb734553d.mp4"
}
```

Key	Description	Type
uploadPath	path to the uploaded video (as a URL that can be accessed from a web browser)	String

Uploading an arbitrary file

[request]

Send some arbitrary files to your system via the URL that was configured as `'editor.upload.file.api'` property.

```
Request URL: http://hostname/uploadFile
Request Method: POST
```

Key	Description	Required	Type
file	file to be uploaded (multipart/form-data)	O	Binary

[response]

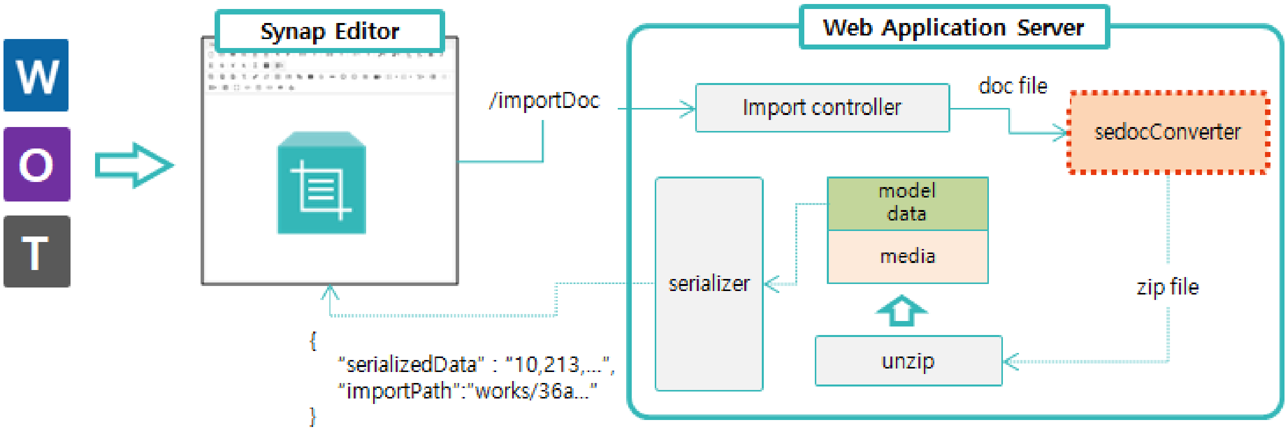
Once the file has been successfully uploaded, your system should respond with the following data:

```
{
    "uploadPath":"uploads/36a43f36f442b5824c6b061eb734553d.docx"
}
```

Key	Description	Required
uploadPath	path to the uploaded file (as a URL that can be accessed from a web browser)	String

Import Document

When you import a document, the format conversion is done at the server side by '`sedocConverter`' module.
'`sedocConverter`' module runs on either Linux or Windows platform, and requires appropriate license to activate.



After choosing the document to be imported, the Editor sends it to the location designated with '`editor.import.api`' key upon configuration (e.g. `/importDoc`).
The service code for that location (e.g. `/importDoc`) which is wrapping around '`sedocConverter`' , needs to be prepared at the customer side, since the service environment differs from site to site.
Please refer to the link below for a sample implementation.

- Java Spring Framework Example
- Java Servlet Example
- ASP.NET (C#) Example
- ASP(Classic) Example
- PHP Example
- PHP4 Example
- Django Example
- Ruby On Rails Example
- Wordpress plugin Example

[request]

Send document file to be imported via the URL that was configured as '`editor.import.api`' property.

```
Request URL: http://hostname/importDoc
Request Method: POST
```

Key	Description	Required	Type
file	document file to be imported (multipart/form-data)	O	Binary

[response]

Once the file has been successfully converted, your system should respond with the following data:

```
{
    "serializedData":[10, 213, 156, ...],
    "importPath": "works/36a43f36f442b5824c6b061eb734553d"
}
```

Key	Description	Type
serializedData	serialized document model data that can be consumed by Editor front-end	Array
importPath	path to the directory where the converted files reside	String