

# Apache Sling

## 1 HTTP Request

Method: GET  
 Path: /wiki/Sling  
 Selector: edit.html  
 Extension: .html  
 Suffix: /richtext?  
 Query Parameters: simple=true  
 HTTP/1.1

## 2 Content Resolution

/wiki/Sling  
 Node has properties

Property	Value
title	Sling Intro
body	<div>Hello World</div>
sling:resourceType	wiki/page

## 3 Get Resource Type

use: sling:resourceType= wiki/page  
 Resource Type  
 fallback: sling:resourceSuperType= «null»  
 last resort: jcr:primaryType= nt:file

## 4 Script Locations

either: /apps/wiki/page/ or: /libs/wiki/page/

## 5 Script Names

Best match: edit.html.esp (Selector+Extension)  
 edit.esp (Selector)  
 html.esp (Extension)  
 Worst match: GET.esp (Method)

## 6 Script

```
<% log.info("Executing my script");
if (request.getRequestParameter("simple")) {
    response.sendRedirect("http://localhost/");
} %>
<html>
<head><title><%=currentNode.title %></title></head>
<body>
<h1><%=resource.getPath() %></h1>
<% out.println(reader.toString()); %>
<% sling.include(resource.getPath() + "content",
    "replaceSelectors= edit"); %>
</body>
</html>
```

## 7 Include Options

```
sling.include("path",
    "forceResourceType= wiki/body, (Powerful)
    replaceSuffix= xhtml,
    addSelectors= foo.bar");
```

# Using the SlingPostServlet this is the default handler for your POST requests. It can do nearly anything.

```
<form action="/mynode" method="POST">  
  <input type="text" name="title">  
  <textarea name="body">  
</form>
```

Create or update /mynode, set title and body. Set lastModified and lastModifiedBy automatically

```
<form action="/mynode/" method="POST">  
  <input type="text" name="dummy">  
  <input type="hidden" name=":order"  
    value="first">  
</form>
```

Create new node below /mynode and make it the first child (also valid: last, before x, after x, 3, 7, 9).

```
<form action="/node" method="POST">  
  <input name=":operation"  
    type="hidden" value="delete">  
</form>
```

Delete /node

```
<form action="/node" method="POST">  
  <input type="hidden"  
    name=":operation" value="delete">  
  <input type="hidden"  
    name=":applyTo" value="/node/one">  
  <input type="hidden"  
    name=":applyTo" value="/node/two">  
</form>
```

Delete /node/one and /node/two

```
<form action="/mynode/" method="POST">  
  <input type="hidden"  
    name=":name" value="new_node">  
  <input type="hidden"  
    name=":nameHint" value="new node">  
</form>
```

Create new node below /mynode, use name or name hint. Set created and createdBy automatically

```
<input type="text" name="customer">  
<input type="hidden" value="John Doe"  
  name="customer@DefaultValue" >  
<input type="hidden" name="title@Delete">
```

Take default value for customer property, remove the title property

```
<form action="/old/node" method="POST">  
  <input type="hidden"  
    name=":operation" value="move">  
  <input type="hidden"  
    name=":dest" value="/new/place">  
</form>
```

Move /old/node to /new/place

```
<input type="text" name="date1"  
  value="2008-06-13T18:55:00">  
<input type="text" name="date2">  
<input type="hidden"  
  name="date2@TypeHint" value="Date">  
<input type="hidden" value="nt:file"  
  name="./uploaded/jcr:primaryType">
```

Guess property type from date pattern, set property type explicitly and set node type explicitly

```
<form action="/old/node" method="POST">  
  <input type="hidden" name=":operation"  
    value="copy">  
  <input type="hidden" name=":dest"  
    value="/new/place">  
  <input type="hidden" name=":replace"  
    value="true">  
</form>
```

Copy /old/node to /new/place and replace the existing node there.

```
<input type="text" name="oldtitle">  
<input type="hidden" value="oldtitle"  
  name="newtitle@ValueFrom">
```

Get value for property title from field oldtitle

```
<input type="hidden" value="/node/prop"  
  name="title@CopyFrom">
```

Copy property title from other node's property