

## Documentation

[Main](#)  
[Getting Started](#)  
[The Sling Engine](#)  
[Development](#)  
[Bundles](#)  
[Tutorials & How-Tos](#)  
[Maven Plugins](#)  
[Configuration](#)

## API Docs

[Sling 11](#)  
[Sling 10](#)  
[Sling 9](#)  
[All versions](#)

## Support

[Wiki](#)  
[FAQ](#)  
[Site Map](#)

## Project Info

[Downloads](#)  
[License](#)  
[News](#)  
[Releases](#)  
[Issue Tracker](#)  
[Links](#)  
[Contributing](#)  
[Project Information](#)  
[Security](#)

## Source

[GitHub](#)  
[Git at Apache](#)

## Apache Software Foundation

[Thanks!](#)  
[Become a Sponsor](#)  
[Buy Stuff](#)



[Home](#) » [Documentation](#) » [The Sling Engine](#) »

[core](#) | [urls](#)

# URL decomposition

- [Overview](#)
- [Examples](#)
- [Automated Tests](#)

## Overview

During the *Resource Resolution* step, the client request URI (as being returned by `[HttpServletRequest.getRequestURI()]`

(`http://docs.oracle.com/javaee/6/api/javax/servlet/http/HttpServletRequest.html#getRequestURI()`) is decomposed into the following parts (in exactly this order):

1. **Resource Path** - For existing resources the resource path is the longest match (also considering its [mappings](#)) pointing to a resource where the next character is either a dot (.) or it is the full request URI. Otherwise (for a path not matching any existing resource) the resource path ends at the *first dot* (.) in the request url. The exact logic for retrieving the resource path is implemented at [ResourceResolver.resolve\(HttpServletRequest,String\)](#). *It is impossible to tell from just looking at the request URI where the resource path part ends. You have to know the underlying resource structure to know how a URL is decomposed. You cannot safely assume that the resource path will always end at the first dot!.*
2. **Selectors** - If the first character in the request URL after the resource path is a dot (.), the string after the dot up to but not including the last dot before the next slash character or the end of the request URL comprises the selectors. If the resource path spans the complete request URL no selectors exist. If only one dot follows the resource path before the end of the request URL or the next slash, also no selectors exist.
3. **Extension** - The string after the last dot after the resource path in the request URL but before the end of the request URL or the next slash after the resource path in the request URL is the extension.
4. **Suffix** - If the request URL contains a slash character after the resource path and optional selectors and extension, the path starting with the slash up to the end of the request URL is the suffix path. Otherwise, the suffix path is empty. Note, that after the resource path at least a dot must be in the URL to let Sling detect the suffix.

Those decomposed parts can be accessed through the `RequestPathInfo` object, which is retrieved via [SlingHttpServletRequest.getPathInfo\(\)](#).

There's a cheat sheet in Adobe's AEM documentation at <https://docs.adobe.com/docs/en/aem/6-2/develop/platform/sling-cheatsheet.html> available to get you familiar with the URL decomposition of Sling.

## Examples

Assume there is a Resource at `/a/b`, which has no children.

URI	Resource Path	Selectors	Extension	Suffix	Resource Found
<code>/a/b</code>	<code>/a/b</code>	null	null	null	yes
<code>/a/b.html</code>	<code>/a/b</code>	null	html	null	yes
<code>/a/b.s1.html</code>	<code>/a/b</code>	s1	html	null	yes
<code>/a/b.s1.s2.html</code>	<code>/a/b</code>	s1.s2	html	null	yes
<code>/a/b/c/d</code>	<code>/a/b/c/d</code>	null	null	null	no!
<code>/a/c.html/s.txt</code>	<code>/a/c</code>	null	html	/s.txt	no!
<code>/a/b./c/d</code>	<code>/a/b</code>	null	null	/c/d	yes
<code>/a/b.html/c/d</code>	<code>/a/b</code>	null	html	/c/d	yes
<code>/a/b.s1.html/c/d</code>	<code>/a/b</code>	s1	html	/c/d	yes
<code>/a/b.s1.s2.html/c/d</code>	<code>/a/b</code>	s1.s2	html	/c/d	yes
<code>/a/b/c/d.s.txt</code>	<code>/a/b/c/d</code>	s	txt	null	no!
<code>/a/b.html/c/d.s.txt</code>	<code>/a/b</code>	null	html	/c/d.s.txt	yes
<code>/a/b.s1.html/c/d.s.txt</code>	<code>/a/b</code>	s1	html	/c/d.s.txt	yes
<code>/a/b.s1.s2.html/c/d.s.txt</code>	<code>/a/b</code>	s1.s2	html	/c/d.s.txt	yes

---

# Automated Tests

---

The tests at

- [ResourceResolverImplTest](#) shows the split between resource path and the rest. Mostly in the method `testBasicAPIAssumptions`.
- [SlingRequestPathInfoTest](#) demonstrates the decomposition after the resource path part.

Feel free to suggest additional tests that help clarify how this works!

*Last modified by Konrad Windszus on Fri Jul 13 11:08:10 2018 +0200*

*Apache Sling, Sling, Apache, the Apache feather logo, and the Apache Sling project logo are trademarks of The Apache Software Foundation. All other marks mentioned may be trademarks or registered trademarks of their respective owners.*

*Copyright © 2007-2018 The Apache Software Foundation.*