

## 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](#) »

[api](#) [core](#)

# Sling API CRUD Support

- [Apache Sling API Support](#)
- [Comparing Sling API CRUD to Sling Post Servlet](#)
  - [Updating a Property](#)
  - [Create New Resource](#)
  - [Remove a Property](#)
  - [Copy a Resource](#)
  - [Move a Resource](#)
  - [Setting non-String Value](#)
  - [Delete a Resource](#)
- [Value Class Support](#)

## Apache Sling API Support

As of version 2.3.0, the Sling API provides full Create Read Update Delete (CRUD) features. CRUD support is provided by the addition of the following methods to the ResourceResolver:

- [void delete\(Resource resource\) throws PersistenceException](#)
- [Resource create\(Resource parent, String name, Map<String, Object> properties\) throws PersistenceException](#)
- [void revert\(\)](#)
- [void commit\(\) throws PersistenceException](#)
- [boolean hasChanges\(\)](#)
- [void refresh\(\)](#)

Which provide the ability to create and delete resources as well as the addition of the ModifiableValueMap interface which is similar to the ValueMap interface, but allows for updating properties on a resource.

## Comparing Sling API CRUD to Sling Post Servlet

Here are some examples of common operations performed using the Sling Post Servlet and Sling API CRUD support. Note, the examples are derived from the [SlingPostServlet Cheatsheet](#).

### Updating a Property

Update /myresource, setting the title and body:

#### Sling Post Servlet

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

#### Sling API CRUD

```
Resource myResource = resourceResolver.getResource("/myresource");
ModifiableValueMap properties = myNode.adaptTo(ModifiableValueMap.class);
properties.put("title", {TITLE});
properties.put("body", {BODY});
resourceResolver.commit();
```

### Create New Resource

Create a new resource below /myresource

#### Sling Post Servlet

```
<form action="/myresource/" method="POST">
  <input type="text" name="dummy">
</form>
```

### Sling API CRUD

```
Resource myResource = resourceResolver.getResource("/myresource");
Map<String,Object> properties = new HashMap<String,Object>();
properties.put("jcr:primaryType", "nt:unstructured");
properties.put("sling:resourceType", "myapp/components/mytype");
Resource dummy = resourceResolver.create(myResource, "dummy", properties);
resourceResolver.commit();
```

## Remove a Property

Remove the property title

### Sling Post Servlet

```
<form action="/myresource" method="POST">
  <input type="hidden" name="title@Delete">
</form>
```

### Sling API CRUD

```
Resource myResource = resourceResolver.getResource("/myresource");
ModifiableValueMap properties = myResource.adaptTo(ModifiableValueMap.class);
properties.remove("title");
resourceResolver.commit();
```

## Copy a Resource

Copy the resource /myresource to /myresource2

### Sling Post Servlet

```
<form action="/myresource" method="POST">
  <input type="hidden" name=":operation" value="copy">
  <input type="hidden" name=":dest" value="/myresource2">
  <input type="hidden" name=":replace" value="true">
</form>
```

### Sling API CRUD

```
Map<String,Object> properties = myResource.adaptTo(ValueMap.class);
Resource myResource2 = resourceResolver.create(null, "myresource2", properties);
resourceResolver.commit();
```

## Move a Resource

Move the resource /myresource2 to /myresource3

### Sling Post Servlet

```
<form action="/myresource2" method="POST">
  <input type="hidden" name=":operation" value="move">
  <input type="hidden" name=":dest" value="/myresource3">
</form>
```

### Sling API CRUD

```
Resource myResource2 = resourceResolver.getResource("/myresource2");
Map<String,Object> properties = myResource2.adaptTo(ValueMap.class);
Resource myResource3 = resourceResolver.create(null, "myresource3", properties);
resourceResolver.delete(myResource2);
resourceResolver.commit();
```

## Setting non-String Value

Set the property date to a particular date

### Sling Post Servlet

```
<form action="/myresource3" method="POST">
  <input type="text" name="date" value="2008-06-13T18:55:00">
  <input type="hidden" name="date@TypeHint" value="Date">
</form>
```

### Sling API CRUD

```
Resource myResource3 = resourceResolver.getResource("/myresource3");
Calendar calendar = [SOME_DATE];
ModifiableValueMap properties = myResource3.adaptTo(ModifiableValueMap.class);
properties.put("date", calendar);
resourceResolver.commit();
```

## Delete a Resource

Delete the resource /myresource

### Sling Post Servlet


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

### Sling API CRUD

```
Resource myResource = resourceResolver.getResource("/myresource");
resourceResolver.delete(myResource);
resourceResolver.commit();
```

---

## Value Class Support

 Please note, this information is specific to the Sling JCR Resource implementation provided by the Apache Sling project. Other implementations may have different value class support.

The Classes implementing the following types are supported directly when setting properties:

- [Calendar](#)
- [InputStream](#)
- [Node](#)
- [BigDecimal](#)
- [Long](#)
- [Short](#)
- [Integer](#)
- [Number](#)
- [Boolean](#)
- [String](#)

As well as the corresponding primitive types. Any object which implements the Serializable interface will be serialized and the result of the serialization will be saved as a binary value for the property.

*Last modified by Bertrand Delacretaz on Fri Sep 29 15:36:08 2017 +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.*