

API

- API
 - WidgetAPI.serverRequest doRequest
 - CodeWidgetActionRequest
 - - String getAction()
 - String getParameter(String parameterName)
 - <T> getParameterAsObject(String parameterName, Class<T> clazz)
 - <T> getParameterAsObject(String parameter, Type type)
 - CodeWidgetResponse
 -
 - WidgetAPI
 - addData(String parameter, Object data)
 - WidgetAPI
 - WidgetAPI.serverRequest(action, actionData)
 -
 -
 -

API

Yellowfin

WidgetAPI.serverRequest doRequest

Yellowfin WidgetAPI.serverRequestJava class

myWidget.js

```
let widgetAPI = options.api.widget;
widgetAPI.serverRequest('hello').then(result => {
  alert(result.hello);
});
```

MyWidgetImplementation.java

```
public void doRequest(CodeWidgetActionRequest request, CodeWidgetActionResponse response) {
    if("hello".equals(request.getAction())) {
        response.addData("hello", "Hello JavaScript!");
    }
}
```

CodeWidgetActionRequest

CodeWidgetActionRequest

String getAction()

WidgetAPI.serverRequest

```
widgetAPI.serverRequest("hello");
```

getAction()hello

String getParameter(String parameterName)

JavaScript actionDataparameterName=null

```
widgetAPI.serverRequest("hello", { name: "Code Widget" });
```

request.getParameter("name") Code Widget

<T> getParameterAsObject(String parameterName, Class<T> clazz)

parameterNameClass

```
Integer counter = getParameterAsObject("counter", Integer.class);
```

null

```
WidgetAPI.serverRequest('example', {
    counter: 1,
    myClass: {
        'one': 'one',
        'two': 'two'
    }
});
```

Java

```
class MyClass {
    private String one = null;
    private String two = null;
    public MyClass(String one, String two) {
        this.one = one;
        this.two = two;
    }
    public String getOne() {
        return one;
    }
    public String getTwo() {
        return two;
    }
}
Integer counter = request.getParameterAsObject('counter', Integer.class);
MyClass myClass = request.getParameterAsObject('myClass', MyClass.class);
System.out.println(myClass.getTwo()); //Prints "two"
```

MyClassJSON

getParameterAsObject(String, Type)

<T> getParameterAsObject(String parameter, Type type)

<T> getParameterAsObject(String parameterName, Class<T> clazz)class

```
List<String> myList = getParameterAsObject("myList", new TypeToken<List<String>>() {}.getType());
```



Yellowfingsonwiki

gson

CodeWidgetResponse

JavaScript

JavaScriptJavaScriptJSON

addData(String parameter, Object data)

JavaScript

```
response.addData('greeting', 'Hello World');
```

Java JavaScript

```
widget.serverRequest('greeting').then(result => {  
    alert(result.greeting);  
})
```

Java classgetPropertyis PropertyJava classJSONgetJSON

getPropertyproperty

isAPropertyaProperty

class

```
public Class MyExampleClass {  
    public String getSayHello() {  
        Return "Hello"  
    }  
    public String helloWorld() {  
        return "Hello World";  
    }  
    public boolean isCodeWidget() {  
        return true;  
    }  
}
```

```
response.addData("exampleClass", new MyExampleClass());
```

```
{  
    sayHello: "Hello",  
    codeWidget: true  
}
```

WidgetAPI

APIJavaScript

WidgetAPI.serverRequest(action, actionData)

Promise - Promise

PromiseAPI#CodeWidgetResponseJSON

action - -

actionData - -

myWidget.js

```
define(function() {  
  
    function MyWidget(options) {  
        let widgetAPI = options.apis.widgetAPI;  
        //Trigger a "hello" request to the backend of the widget  
        widgetAPI.serverRequest('hello').then(result => {  
            alert(result.hello);  
        });  
    }  
    return MyWidget;  
}
```

MyWidgetImplementation.java

```
public void doRequest(CodeWidgetActionRequest request, CodeWidgetActionResponse response) {  
  
    if("hello".equals(request.getAction())) {  
        response.addData("hello", "Hello JavaScript!");  
    }  
}
```