

KARLSRUHE INSTITUTE OF TECHNOLOGY

SOFTWARE ENGINEERING PRACTICE

WINTER TERM 2015/2016

# rootJS

Node.js bindings for ROOT 6

*Jonas Schwabe*

*Theo Beffart*

*Sachin Rajgopal*

*Christoph Wolff*

*Christoph Haas*

*Maximilian Früh*

supervised by  
Dr. Marek SZUBA

# Contents

# 1. NodeApplication

describe class NodeApplication here

## 1.1. ctorCallback

<i>Name</i>	<code>NodeApplication::ctorCallback(args: FunctionCallbackInfo&lt;Value&gt;)</code>
<i>Visibility</i>	Public
<i>Parameters</i>	<i>args: FunctionCallbackInfo&lt;Value&gt;</i>
<i>Return value</i>	<b>none</b>
<i>behavior</i>	describe beahviour

## 1.2. staticCtorCallback

<i>Name</i>	<code>NodeApplication::staticCtorCallback(args: FunctionCallbackInfo&lt;Value&gt;)</code>
<i>Visibility</i>	Public
<i>Parameters</i>	<i>args: FunctionCallbackInfo&lt;Value&gt;</i>
<i>Return value</i>	<b>none</b>
<i>behavior</i>	describe beahviour

### 1.3. memberGetterCallback

<i>Name</i>	<code>NodeApplication::memberGetterCallback(property: Local&lt;String&gt;, info: PropertyCallbackInfo&lt;Value&gt;)</code>
<i>Visibility</i>	Public
<i>Parameters</i>	<code>property: Local&lt;String&gt;, info: PropertyCallbackInfo&lt;Value&gt;</code>
<i>Return value</i>	<b>none</b>
<i>behavior</i>	describe beahviour

## 1.4. memberSetterCallback

<i>Name</i>	<code>NodeApplication::memberSetterCallback(property: Local&lt;String&gt;, value: Local&lt;Value&gt;, info: PropertyCallbackInfo&lt;Value&gt;)</code>
<i>Visibility</i>	Public
<i>Parameters</i>	<code>property: Local&lt;String&gt;, value: Local&lt;Value&gt;, info: PropertyCallbackInfo&lt;Value&gt;</code>
<i>Return value</i>	<b>none</b>
<i>behavior</i>	describe beahviour

## 1.5. memberFunctionCallback

<i>Name</i>	<code>NodeApplication::memberFunctionCallback(args: FunctionCallbackInfo&lt;Value&gt;)</code>
<i>Visibility</i>	Public
<i>Parameters</i>	<i>args: FunctionCallbackInfo&lt;Value&gt;</i>
<i>Return value</i>	<b>none</b>
<i>behavior</i>	describe beahviour

## 1.6. staticGetterCallback

<i>Name</i>	<code>NodeApplication::staticGetterCallback(property: Local&lt;String&gt;, info: PropertyCallbackInfo&lt;Value&gt;)</code>
<i>Visibility</i>	Public
<i>Parameters</i>	<code>property: Local&lt;String&gt;, info: PropertyCallbackInfo&lt;Value&gt;</code>
<i>Return value</i>	<b>none</b>
<i>behavior</i>	describe beahviour



## 1.7. staticSetterCallback

<i>Name</i>	<code>NodeApplication::staticSetterCallback(property: Local&lt;String&gt;, value: Local&lt;Value&gt;, info: PropertyCallbackInfo&lt;Value&gt;)</code>
<i>Visibility</i>	Public
<i>Parameters</i>	<i>property:</i> <code>Local&lt;String&gt;</code> , <i>value:</i> <code>Local&lt;Value&gt;</code> , <i>info:</i> <code>PropertyCallbackInfo&lt;Value&gt;</code>
<i>Return value</i>	<b>none</b>
<i>behavior</i>	describe beahviour

## 1.8. staticFunctionCallback

<i>Name</i>	<code>NodeApplication::staticFunctionCallback(args: FunctionCallbackInfo&lt;Value&gt;)</code>
<i>Visibility</i>	Public
<i>Parameters</i>	<i>args: FunctionCallbackInfo&lt;Value&gt;</i>
<i>Return value</i>	<b>none</b>
<i>behavior</i>	describe beahviour

## 1.9. Initialize

<i>Name</i>	<code>NodeApplication::Initialize(exports: Local&lt;Object&gt;, module: Local&lt;Object&gt;)</code>
<i>Visibility</i>	Public
<i>Parameters</i>	<i>exports: Local&lt;Object&gt;, module: Local&lt;Object&gt;</i>
<i>Return value</i>	<b>none</b>
<i>behavior</i>	describe beahviour

## 1.10. Instance

<i>Name</i>	<code>NodeApplication::Instance()</code>
<i>Visibility</i>	Public
<i>Parameters</i>	<i>none</i>
<i>Return value</i>	<b>NodeApplication</b> describe return value
<i>behavior</i>	describe beahviour

### 1.11. getIsolate

<i>Name</i>	<code>NodeApplication::getIsolate()</code>
<i>Visibility</i>	Public
<i>Parameters</i>	<i>none</i>
<i>Return value</i>	<b>Isolate*</b> describe return value
<i>behavior</i>	describe beahviour

## 1.12. getExports

<i>Name</i>	<code>NodeApplication::getExports()</code>
<i>Visibility</i>	Public
<i>Parameters</i>	<i>none</i>
<i>Return value</i>	<b>Local</b> < <b>Object</b> > describe return value
<i>behavior</i>	describe beahviour

### 1.13. getTemplateFactory

<i>Name</i>	NodeApplication::getTemplateFactory()
<i>Visibility</i>	Public
<i>Parameters</i>	<i>none</i>
<i>Return value</i>	<b>TemplateFactory</b> describe return value
<i>behavior</i>	describe beahviour

## 1.14. getFunctionFactory

<i>Name</i>	<code>NodeApplication::getFunctionFactory()</code>
<i>Visibility</i>	Public
<i>Parameters</i>	<i>none</i>
<i>Return value</i>	<b>ProxyFunctionFactory</b> describe return value
<i>behavior</i>	describe beahviour



## 1.15. getObjectFactory

<i>Name</i>	<code>NodeApplication::getObjectFactory()</code>
<i>Visibility</i>	Public
<i>Parameters</i>	<i>none</i>
<i>Return value</i>	<b>ProxyObjectFactory</b> describe return value
<i>behavior</i>	describe beahviour

## 2. TemplateFactory

describe class TemplateFactory here

### 2.1. createTemplate

<i>Name</i>	TemplateFactory::createTemplate(clazz: TClassRef)
<i>Visibility</i>	Public
<i>Parameters</i>	clazz: TClassRef
<i>Return value</i>	<b>Local&lt;FunctionTemplate&gt;</b> describe return value
<i>behavior</i>	describe beahviour

## 3. TemplateCache

describe class TemplateCache here

### 3.1. contains

<i>Name</i>	TemplateCache::contains(type: TClassRef)
<i>Visibility</i>	Public
<i>Parameters</i>	<i>type: TClassRef</i>
<i>Return value</i>	<b>bool</b> describe return value
<i>behavior</i>	describe beahviour

### 3.2. get

<i>Name</i>	<code>TemplateCache::get(type: TClassRef)</code>
<i>Visibility</i>	Public
<i>Parameters</i>	<i>type: TClassRef</i>
<i>Return value</i>	<b>Local&lt;FunctionTemplate&gt;</b> describe return value
<i>behavior</i>	describe beahviour

### 3.3. store

<i>Name</i>	<code>TemplateCache::store(type: TClassRef, tpl: Local&lt;FunctionTemplate&gt;)</code>
<i>Visibility</i>	Public
<i>Parameters</i>	<i>type: TClassRef, tpl: Local&lt;FunctionTemplate&gt;</i>
<i>Return value</i>	<b>none</b>
<i>behavior</i>	describe beahviour

## 4. ProxyFunctionFactory

describe class ProxyFunctionFactory here

### 4.1. createProxyFunction

<i>Name</i>	ProxyFunctionFactory::createProxyFunction(info: TMethod)
<i>Visibility</i>	Public
<i>Parameters</i>	<i>info: TMethod</i>
<i>Return value</i>	<b>ProxyFunciton</b> describe return value
<i>behavior</i>	describe beahviour

## 4.2. fromArgs

<i>Name</i>	ProxyFunctionFactory::fromArgs(name: string, clazz: TClassRef, args: FunctionCallbackInfo)
<i>Visibility</i>	Public
<i>Parameters</i>	name: string, clazz: TClassRef, args: FunctionCallbackInfo
<i>Return value</i>	<b>ProxyFunction</b> describe return value
<i>behavior</i>	describe beahviour

## 5. ProxyFunction

describe class ProxyFunction here

### 5.1. ProxyFunction

<i>Name</i>	ProxyFunction::ProxyFunction(address: void*, info: TFunction)
<i>Visibility</i>	Public
<i>Parameters</i>	address: void*, info: TFunction
<i>Return value</i>	« <b>constructor</b> » describe return value
<i>behavior</i>	describe beahviour



## 5.2. convertArgs

<i>Name</i>	<code>ProxyFunction::convertArgs(args: FunctionCallbackInfo)</code>
<i>Visibility</i>	Public
<i>Parameters</i>	<i>args: FunctionCallbackInfo</i>
<i>Return value</i>	<b>ProxyObject[]</b> describe return value
<i>behavior</i>	describe beahviour

### 5.3. call

<i>Name</i>	<code>ProxyFunction::call(args: ProxyObject[])</code>
<i>Visibility</i>	Public
<i>Parameters</i>	<i>args: ProxyObject[]</i>
<i>Return value</i>	<b>ProxyObject</b> describe return value
<i>behavior</i>	describe beahviour

## 5.4. isTemplateFunction

<i>Name</i>	ProxyFunction::isTemplateFunction()
<i>Visibility</i>	Public
<i>Parameters</i>	<i>none</i>
<i>Return value</i>	<b>bool</b> describe return value
<i>behavior</i>	describe beahviour

## 6. ProxyObjectFactory

describe class ProxyObjectFactory here

### 6.1. createProxyObject

<i>Name</i>	ProxyObjectFactory::createProxyObject(type: TDataMember, holder: ProxyObject)
<i>Visibility</i>	Public
<i>Parameters</i>	<i>type: TDataMember, holder: ProxyObject</i>
<i>Return value</i>	<b>ProxyObject</b> describe return value
<i>behavior</i>	describe beahviour

## 7. ProxyObject

describe class ProxyObject here

### 7.1. ProxyObject

<i>Name</i>	ProxyObject::ProxyObject(address: void*, type: TDataMember)
<i>Visibility</i>	Public
<i>Parameters</i>	address: void*, type: TDataMember
<i>Return value</i>	« <b>constructor</b> » describe return value
<i>behavior</i>	describe beahviour

## 7.2. getAddress

<i>Name</i>	<code>ProxyObject::getAddress()</code>
<i>Visibility</i>	Public
<i>Parameters</i>	<i>none</i>
<i>Return value</i>	<b>void*</b> describe return value
<i>behavior</i>	describe beahviour

### 7.3. getType

<i>Name</i>	<code>ProxyObject::getType()</code>
<i>Visibility</i>	Public
<i>Parameters</i>	<i>none</i>
<i>Return value</i>	<b>TDataMember</b> describe return value
<i>behavior</i>	describe beahviour

## 7.4. set

<i>Name</i>	<code>ProxyObject::set(value: ProxyObject)</code>
<i>Visibility</i>	Public
<i>Parameters</i>	<i>value: ProxyObject</i>
<i>Return value</i>	<b>none</b>
<i>behavior</i>	describe beahviour



## 7.5. get

<i>Name</i>	<code>ProxyObject::get()</code>
<i>Visibility</i>	Public
<i>Parameters</i>	<i>none</i>
<i>Return value</i>	<b>Local</b> <Value> describe return value
<i>behavior</i>	describe beahviour

## 7.6. isPrimitive

<i>Name</i>	<code>ProxyObject::isPrimitive()</code>
<i>Visibility</i>	Public
<i>Parameters</i>	<i>none</i>
<i>Return value</i>	<b>bool</b> describe return value
<i>behavior</i>	describe beahviour