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. CallbackHandler

describe class CallbackHandler here

1.1. ctorCallback

Name	`CallbackHandler::ctorCallback(args: FunctionCallbackInfo<Value>)`
Visibility	public
Parameters	*args: FunctionCallbackInfo<Value>*
Return value	**none**
behavior	describe beahviour

1.2. staticCtorCallback

Name	`CallbackHandler::staticCtorCallback(args: FunctionCallbackInfo<Value>)`
Visibility	public
Parameters	*args: FunctionCallbackInfo<Value>*
Return value	**none**
behavior	describe beahviour

1.3. memberGetterCallback

Name	`CallbackHandler::memberGetterCallback(property: Local<String>, info: PropertyCallbackInfo<Value>)`
Visibility	public
Parameters	*property: Local<String>, info: PropertyCallbackInfo<Value>*
Return value	**none**
behavior	describe beahviour

1.4. memberSetterCallback

Name	`CallbackHandler::memberSetterCallback(property: Local<String>, value: Local<Value>, info: PropertyCallbackInfo<Value>)`
Visibility	public
Parameters	*property: Local<String>, value: Local<Value>, info: PropertyCallbackInfo<Value>*
Return value	**none**
behavior	describe beahviour

1.5. memberFunctionCallback

Name	`CallbackHandler::memberFunctionCallback(args: FunctionCallbackInfo<Value>)`
Visibility	public
Parameters	*args: FunctionCallbackInfo<Value>*
Return value	**none**
behavior	describe beahviour

1.6. staticGetterCallback

Name	`CallbackHandler::staticGetterCallback(property: Local<String>, info: PropertyCallbackInfo<Value>)`
Visibility	public
Parameters	*property: Local<String>, info: PropertyCallbackInfo<Value>*
Return value	**none**
behavior	describe beahviour

1.7. staticSetterCallback

Name	`CallbackHandler::staticSetterCallback(property: Local<String>, value: Local<Value>, info: PropertyCallbackInfo<Value>)`
Visibility	public
Parameters	*property: Local<String>, value: Local<Value>, info: PropertyCallbackInfo<Value>*
Return value	**none**
behavior	describe beahviour

1.8. staticFunctionCallback

Name	`CallbackHandler::staticFunctionCallback(args: FunctionCallbackInfo<Value>)`
Visibility	public
Parameters	*args: FunctionCallbackInfo<Value>*
Return value	**none**
behavior	describe beahviour

2. NodeHandler

describe class NodeHandler here

2.1. getExports

Name	`NodeHandler::getExports()`
Visibility	public
Parameters	*none*
Return value	**Local**<**Object**> describe return value
behavior	describe beahviour

3. NodeApplication

describe class NodeApplication here

3.1. NodeApplication

Name	`NodeApplication::NodeApplication(acn: char*, argc: int*, argv: char**)`
Visibility	public
Parameters	*acn: char*, argc: int*, argv: char***
Return value	**«constructor»** describe return value
behavior	describe beahviour

4. TemplateFactory

describe class TemplateFactory here

4.1. createTemplate

Name	`TemplateFactory::createTemplate(clazz: TClassRef)`
Visibility	public
Parameters	*clazz: TClassRef*
Return value	**Local**<**FunctionTemplate**> describe return value
behavior	describe beahviour

5. Proxy

describe class Proxy here

5.1. Proxy

Name	`Proxy::Proxy(address: void*, type: TObject, scope: TClassRef)`
Visibility	protected
Parameters	*address: void*, type: TObject, scope: TClassRef*
Return value	**«constructor»** describe return value
behavior	describe beahviour

5.2. setAddress

Name	`Proxy::setAddress(address: void*)`
Visibility	public
Parameters	*address: void**
Return value	**none**
behavior	describe beahviour

5.3. getAddress

Name	`Proxy::getAddress()`
Visibility	public
Parameters	*none*
Return value	**void*** describe return value
behavior	describe beahviour

5.4. getType

Name	`Proxy::getType()`
Visibility	public
Parameters	*none*
Return value	**TObject** describe return value
behavior	describe beahviour

5.5. getScope

Name	`Proxy::getScope()`
Visibility	public
Parameters	*none*
Return value	**TClassRef** describe return value
behavior	describe beahviour

5.6. isGlobal

Name	`Proxy::isGlobal()`
Visibility	public
Parameters	*none*
Return value	**bool** describe return value
behavior	describe beahviour

5.7. isTemplate

Name	`Proxy::isTemplate()`
Visibility	public
Parameters	*none*
Return value	**bool** describe return value
behavior	describe beahviour

5.8. isConst

Name	`Proxy::isConst()`
Visibility	public
Parameters	*none*
Return value	**bool** describe return value
behavior	describe beahviour

5.9. isStatic

Name	`Proxy::isStatic()`
Visibility	public
Parameters	*none*
Return value	**bool** describe return value
behavior	describe beahviour

6. ProxyFunctionFactory

describe class ProxyFunctionFactory here

6.1. createProxyFunction

Name	`ProxyFunctionFactory::createProxyFunction(function: TFunction, scope: TClassRef)`
Visibility	public
Parameters	*function: TFunction, scope: TClassRef*
Return value	**ProxyFunciton** describe return value
behavior	describe beahviour

6.2. fromArgs

Name	`ProxyFunctionFactory::fromArgs(name: string, scope: TClassRef, args: FunctionCallbackInfo)`
Visibility	public
Parameters	*name: string, scope: TClassRef, args: FunctionCallbackInfo*
Return value	**ProxyFunction** describe return value
behavior	describe beahviour

7. ProxyFunction

describe class ProxyFunction here

7.1. getCallFunc

Name	`ProxyFunction::getCallFunc(method: TFunction*)`
Visibility	public
Parameters	*method: TFunction**
Return value	**CallFunc*** describe return value
behavior	describe beahviour

7.2. getMethodsFromName

Name	`ProxyFunction::getMethodsFromName(scope: TClassRef, name: string)`
Visibility	public
Parameters	*scope: TClassRef, name: string*
Return value	**vector<TFunction*>** describe return value
behavior	describe beahviour

7.3. ProxyFunction

Name	`ProxyFunction::ProxyFunction(address: void*, function: TFunction, scope: TClassRef)`
Visibility	public
Parameters	*address: void*, function: TFunction, scope: TClassRef*
Return value	**«constructor»** describe return value
behavior	describe beahviour

7.4. getType

Name	`ProxyFunction::getType()`
Visibility	public
Parameters	*none*
Return value	**TFunction** describe return value
behavior	describe beahviour

7.5. validateArgs

Name	`ProxyFunction::validateArgs(args: FunctionCallbackInfo)`
Visibility	public
Parameters	*args: FunctionCallbackInfo*
Return value	**ProxyObject[]** describe return value
behavior	describe beahviour

7.6. call

Name	`ProxyFunction::call(args: ProxyObject[])`
Visibility	public
Parameters	*args: ProxyObject[]*
Return value	**ProxyObject** describe return value
behavior	describe beahviour

8. ProxyObjectFactory

describe class ProxyObjectFactory here

8.1. createProxyObject

Name	`ProxyObjectFactory::createProxyObject(type: TDataMember, scope: TClassRef, holder: ProxyObject)`
Visibility	public
Parameters	*type: TDataMember, scope: TClassRef, holder: ProxyObject*
Return value	**ProxyObject** describe return value
behavior	describe beahviour

9. ProxyObject

describe class ProxyObject here

9.1. ProxyObject

Name	`ProxyObject::ProxyObject(type: TDataMember, scope: TClassRef)`
Visibility	public
Parameters	*type: TDataMember, scope: TClassRef*
Return value	**«constructor»** describe return value
behavior	describe beahviour

9.2. getType

Name	`ProxyObject::getType()`
Visibility	public
Parameters	*none*
Return value	**TDataMember** describe return value
behavior	describe beahviour

9.3. set

Name	`ProxyObject::set(value: ProxyObject)`
Visibility	public
Parameters	*value: ProxyObject*
Return value	**none**
behavior	describe beahviour

9.4. get

Name	`ProxyObject::get()`
Visibility	public
Parameters	*none*
Return value	**Local**<**Value**> describe return value
behavior	describe beahviour

9.5. setProxy

Name	`ProxyObject::setProxy(proxy: Local<Object>)`
Visibility	public
Parameters	*proxy: Local<Object>*
Return value	**none**
behavior	describe beahviour

9.6. getProxy

Name	`ProxyObject::getProxy()`
Visibility	public
Parameters	*none*
Return value	**Local**<**Object**> describe return value
behavior	describe beahviour

9.7. isPrimitive

Name	`ProxyObject::isPrimitive()`
Visibility	public
Parameters	*none*
Return value	**bool** describe return value
behavior	describe beahviour