

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	3
1.1	ctorCallback	3
1.2	staticCtorCallback	4
1.3	memberGetterCallback	5
1.4	memberSetterCallback	6
1.5	memberFunctionCallback	7
1.6	staticGetterCallback	8
1.7	staticSetterCallback	9
1.8	staticFunctionCallback	10
1.9	Initialize	11
1.10	Instance	12
1.11	getIsolate	13
1.12	getExports	14
1.13	getTemplateFactory	15
1.14	getFunctionFactory	16
1.15	getObjectFactory	17
2	TemplateFactory	18
2.1	TemplateFactory	18
2.2	createTemplate	19
3	Proxy	20
3.1	Proxy	20
3.2	setAddress	21
3.3	getAddress	22
3.4	getType	23
3.5	getScope	24
3.6	isGlobal	25
3.7	isTemplate	26
3.8	isConst	27
3.9	isStatic	28
4	ProxyFunctionFactory	29
4.1	ProxyFunctionFactory	29
4.2	createProxyFunction	30
4.3	fromArgs	31
5	ProxyFunction	32
5.1	getCallFunc	32
5.2	getMethodsFromName	33
5.3	ProxyFunction	34
5.4	getType	35
5.5	validateArgs	36
5.6	call	37
6	ProxyObjectFactory	38
6.1	ProxyObjectFactory	38
6.2	createProxyObject	39

7	ProxyObject	40
7.1	ProxyObject	40
7.2	getType	41
7.3	set	42
7.4	get	43
7.5	setProxy	44
7.6	getProxy	45
7.7	isPrimitive	46
8	TemplateProxy	47
9	EnumProxy	48
10	StructProxy	49
11	ArrayProxy	50
12	PointerProxy	51
13	ReferenceProxy	52
14	PrimitiveProxy	53
15	NumberProxy	54
16	StringProxy	55
17	BooleanProxy	56

1. NodeApplication

describe class NodeApplication here

1.1. ctorCallback

<i>Name</i>	NodeApplication::ctorCallback(args: FunctionCallbackInfo<Value>)
<i>Visibility</i>	public
<i>Parameters</i>	args: FunctionCallbackInfo<Value>
<i>Return value</i>	none
<i>behavior</i>	describe beahviour

1.2. staticCtorCallback

<i>Name</i>	<code>NodeApplication::staticCtorCallback(args: FunctionCallbackInfo<Value>)</code>
<i>Visibility</i>	<code>public</code>
<i>Parameters</i>	<code>args: FunctionCallbackInfo<Value></code>
<i>Return value</i>	<code>none</code>
<i>behavior</i>	describe beahviour

1.3. memberGetterCallback

<i>Name</i>	<code>NodeApplication::memberGetterCallback(property: Local<String>, info: PropertyCallbackInfo<Value>)</code>
<i>Visibility</i>	<code>public</code>
<i>Parameters</i>	<code>property: Local<String>, info: PropertyCallbackInfo<Value></code>
<i>Return value</i>	<code>none</code>
<i>behavior</i>	describe beahviour

1.4. memberSetterCallback

<i>Name</i>	<code>NodeApplication::memberSetterCallback(property: Local<String>, value: Local<Value>, info: PropertyCallbackInfo<Value>)</code>
<i>Visibility</i>	<code>public</code>
<i>Parameters</i>	<code>property: Local<String>, value: Local<Value>, info: PropertyCallbackInfo<Value></code>
<i>Return value</i>	none
<i>behavior</i>	describe beahviour

1.5. memberFunctionCallback

<i>Name</i>	<code>NodeApplication::memberFunctionCallback(args: FunctionCallbackInfo<Value>)</code>
<i>Visibility</i>	public
<i>Parameters</i>	<i>args: FunctionCallbackInfo<Value></i>
<i>Return value</i>	none
<i>behavior</i>	describe beahviour

1.6. staticGetterCallback

<i>Name</i>	<code>NodeApplication::staticGetterCallback(property: Local<String>, info: PropertyCallbackInfo<Value>)</code>
<i>Visibility</i>	<code>public</code>
<i>Parameters</i>	<code>property: Local<String>, info: PropertyCallbackInfo<Value></code>
<i>Return value</i>	<code>none</code>
<i>behavior</i>	describe beahviour

1.7. staticSetterCallback

<i>Name</i>	<code>NodeApplication::staticSetterCallback(property: Local<String>, value: Local<Value>, info: PropertyCallbackInfo<Value>)</code>
<i>Visibility</i>	<code>public</code>
<i>Parameters</i>	<code>property: Local<String>, value: Local<Value>, info: PropertyCallbackInfo<Value></code>
<i>Return value</i>	<code>none</code>
<i>behavior</i>	describe beahviour

1.8. staticFunctionCallback

<i>Name</i>	<code>NodeApplication::staticFunctionCallback(args: FunctionCallbackInfo<Value>)</code>
<i>Visibility</i>	<code>public</code>
<i>Parameters</i>	<code>args: FunctionCallbackInfo<Value></code>
<i>Return value</i>	<code>none</code>
<i>behavior</i>	describe beahviour

1.9. Initialize

<i>Name</i>	<code>NodeApplication::Initialize(exports: Local<Object>, module: Local<Object>)</code>
<i>Visibility</i>	<code>public</code>
<i>Parameters</i>	<code>exports: Local<Object>, module: Local<Object></code>
<i>Return value</i>	<code>none</code>
<i>behavior</i>	describe beahviour

1.10. Instance

<i>Name</i>	<code>NodeApplication::Instance()</code>
<i>Visibility</i>	public
<i>Parameters</i>	none
<i>Return value</i>	NodeApplication 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>	Isolate* 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>	Local<Object> describe return value
<i>behavior</i>	describe beahviour

1.13. getTemplateFactory

<i>Name</i>	<code>NodeApplication::getTemplateFactory()</code>
<i>Visibility</i>	public
<i>Parameters</i>	none
<i>Return value</i>	TemplateFactory 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>	none
<i>Return value</i>	ProxyFunctionFactory 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>	none
<i>Return value</i>	ProxyObjectFactory describe return value
<i>behavior</i>	describe beahviour

2. TemplateFactory

describe class TemplateFactory here

2.1. TemplateFactory

<i>Name</i>	TemplateFactory::TemplateFactory()
<i>Visibility</i>	public
<i>Parameters</i>	none
<i>Return value</i>	« constructor » describe return value
<i>behavior</i>	describe beahviour

2.2. createTemplate

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

3. Proxy

describe class Proxy here

3.1. Proxy

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

3.2. setAddress

<i>Name</i>	<code>Proxy::setAddress(address: void*)</code>
<i>Visibility</i>	<code>public</code>
<i>Parameters</i>	<code>address: void*</code>
<i>Return value</i>	none
<i>behavior</i>	describe beahviour

3.3. getAddress

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

3.4. getType

<i>Name</i>	<code>Proxy::getType()</code>
<i>Visibility</i>	public
<i>Parameters</i>	none
<i>Return value</i>	TObject describe return value
<i>behavior</i>	describe beahviour

3.5. getScope

<i>Name</i>	<code>Proxy::getScope()</code>
<i>Visibility</i>	public
<i>Parameters</i>	none
<i>Return value</i>	TClassRef describe return value
<i>behavior</i>	describe beahviour

3.6. isGlobal

<i>Name</i>	<code>Proxy::isGlobal()</code>
<i>Visibility</i>	public
<i>Parameters</i>	<i>none</i>
<i>Return value</i>	bool describe return value
<i>behavior</i>	describe beahviour

3.7. isTemplate

<i>Name</i>	<code>Proxy::isTemplate()</code>
<i>Visibility</i>	public
<i>Parameters</i>	none
<i>Return value</i>	bool describe return value
<i>behavior</i>	describe beahviour

3.8. isConst

<i>Name</i>	<code>Proxy::isConst()</code>
<i>Visibility</i>	public
<i>Parameters</i>	none
<i>Return value</i>	bool describe return value
<i>behavior</i>	describe beahviour

3.9. isStatic

<i>Name</i>	<code>Proxy::isStatic()</code>
<i>Visibility</i>	public
<i>Parameters</i>	none
<i>Return value</i>	bool describe return value
<i>behavior</i>	describe beahviour

4. ProxyFunctionFactory

describe class ProxyFunctionFactory here

4.1. ProxyFunctionFactory

<i>Name</i>	ProxyFunctionFactory::ProxyFunctionFactory()
<i>Visibility</i>	public
<i>Parameters</i>	none
<i>Return value</i>	« constructor » describe return value
<i>behavior</i>	describe beahviour

4.2. createProxyFunction

<i>Name</i>	<code>ProxyFunctionFactory::createProxyFunction(function: TFunction, scope: TClassRef)</code>
<i>Visibility</i>	public
<i>Parameters</i>	<i>function: TFunction, scope: TClassRef</i>
<i>Return value</i>	ProxyFunciton describe return value
<i>behavior</i>	describe beahviour

4.3. fromArgs

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

5. ProxyFunction

describe class ProxyFunction here

5.1. getCallFunc

<i>Name</i>	ProxyFunction::getCallFunc(method: TFunction*)
<i>Visibility</i>	public
<i>Parameters</i>	method: TFunction*
<i>Return value</i>	CallFunc* describe return value
<i>behavior</i>	describe beahviour

5.2. getMethodsFromName

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

5.3. ProxyFunction

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

5.4. getType

<i>Name</i>	<code>ProxyFunction::getType()</code>
<i>Visibility</i>	public
<i>Parameters</i>	<i>none</i>
<i>Return value</i>	TFunction describe return value
<i>behavior</i>	describe beahviour

5.5. validateArgs

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

5.6. 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>	ProxyObject describe return value
<i>behavior</i>	describe beahviour

6. ProxyObjectFactory

describe class ProxyObjectFactory here

6.1. ProxyObjectFactory

<i>Name</i>	ProxyObjectFactory::ProxyObjectFactory()
<i>Visibility</i>	public
<i>Parameters</i>	none
<i>Return value</i>	« constructor » describe return value
<i>behavior</i>	describe beahviour

6.2. createProxyObject

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

7. ProxyObject

describe class ProxyObject here

7.1. ProxyObject

<i>Name</i>	ProxyObject::ProxyObject(type: TDataMember, scope: TClassRef)
<i>Visibility</i>	public
<i>Parameters</i>	type: TDataMember, scope: TClassRef
<i>Return value</i>	« constructor » describe return value
<i>behavior</i>	describe beahviour

7.2. getType

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

7.3. 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>	none
<i>behavior</i>	describe beahviour

7.4. get

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

7.5. setProxy

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

7.6. getProxy

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

7.7. isPrimitive

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

8. TemplateProxy

describe class TemplateProxy here

9. EnumProxy

describe class EnumProxy here

10. StructProxy

describe class StructProxy here

11. ArrayProxy

describe class ArrayProxy here

12. PointerProxy

describe class PointerProxy here

13. ReferenceProxy

describe class ReferenceProxy here

14. PrimitiveProxy

describe class PrimitiveProxy here

15. NumberProxy

describe class NumberProxy here

16. StringProxy

describe class StringProxy here

17. BooleanProxy

describe class BooleanProxy here