

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	2
1.1	ctorCallback	2
1.2	staticCtorCallback	3
1.3	memberGetterCallback	4
1.4	memberSetterCallback	5
1.5	memberFunctionCallback	6
1.6	staticGetterCallback	7
1.7	staticSetterCallback	8
1.8	staticFunctionCallback	9
1.9	Initialize	10
1.10	Instance	11
1.11	getIsolate	12
1.12	getExports	13
1.13	getTemplateFactory	14
1.14	getFunctionFactory	15
1.15	getObjectFactory	16
2	TemplateFactory	17
2.1	createTemplate	17
3	TemplateCache	18
3.1	contains	18
3.2	get	19
3.3	store	20
4	ProxyFunctionFactory	21
4.1	createProxyFunction	21
4.2	fromArgs	22
5	ProxyFunction	23
5.1	ProxyFunction	23
5.2	convertArgs	24
5.3	call	25
5.4	isTemplateFunction	26
6	ProxyObjectFactory	27
6.1	createProxyObject	27
7	ProxyObject	28
7.1	ProxyObject	28
7.2	getAddress	29
7.3	getType	30
7.4	set	31
7.5	get	32
7.6	isPrimitive	33

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>	Public
<i>Parameters</i>	<i>args: FunctionCallbackInfo<Value></i>
<i>Return value</i>	none
<i>behavior</i>	describe beahviour

1.3. memberGetterCallback

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

1.8. staticFunctionCallback

<i>Name</i>	<code>NodeApplication::staticFunctionCallback(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

1.9. Initialize

<i>Name</i>	<code>NodeApplication::Initialize(exports: Local<Object>, module: Local<Object>)</code>
<i>Visibility</i>	Public
<i>Parameters</i>	<i>exports: Local<Object>, module: Local<Object></i>
<i>Return value</i>	none
<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>	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>	NodeApplication::getTemplateFactory()
<i>Visibility</i>	Public
<i>Parameters</i>	<i>none</i>
<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>	<i>none</i>
<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>	<i>none</i>
<i>Return value</i>	ProxyObjectFactory 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>	Local<FunctionTemplate> 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>	bool 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>	Local<FunctionTemplate> describe return value
<i>behavior</i>	describe beahviour

3.3. store

<i>Name</i>	<code>TemplateCache::store(type: TClassRef, tpl: Local<FunctionTemplate>)</code>
<i>Visibility</i>	Public
<i>Parameters</i>	<i>type: TClassRef, tpl: Local<FunctionTemplate></i>
<i>Return value</i>	none
<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>	ProxyFunciton 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>	ProxyFunction 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>	« constructor » 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>	ProxyObject[] 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>	ProxyObject 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>	bool 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>	ProxyObject 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>	« constructor » 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>	void* 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>	TDataMember 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>	none
<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>	Local <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>	bool describe return value
<i>behavior</i>	describe beahviour