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.tex

describe class NodeApplication.tex here

1.1. ctorCallback

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

1.2. staticCtorCallback

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

1.3. memberGetterCallback

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

1.6. staticGetterCallback

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

1.9. Initialize

Name	`NodeApplication.tex::Initialize(exports: Local<Object>, module: Local<Object>)`
Visibility	public
Parameters	*exports: Local<Object>, module: Local<Object>*
Return value	**none**
behavior	describe beahviour

1.10. Instance

Name	`NodeApplication.tex::Instance()`
Visibility	public
Parameters	*none*
Return value	**NodeApplication** describe return value
behavior	describe beahviour

1.11. getIsolate

Name	`NodeApplication.tex::getIsolate()`
Visibility	public
Parameters	*none*
Return value	**Isolate*** describe return value
behavior	describe beahviour

1.12. getExports

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

1.13. getTemplateFactory

Name	`NodeApplication.tex::getTemplateFactory()`
Visibility	public
Parameters	*none*
Return value	**TemplateFactory** describe return value
behavior	describe beahviour

1.14. getFunctionFactory

Name	`NodeApplication.tex::getFunctionFactory()`
Visibility	public
Parameters	*none*
Return value	**ProxyFunctionFactory** describe return value
behavior	describe beahviour

1.15. getObjectFactory

Name	`NodeApplication.tex::getObjectFactory()`
Visibility	public
Parameters	*none*
Return value	**ProxyObjectFactory** describe return value
behavior	describe beahviour

2. TemplateFactory

describe class TemplateFactory here

2.1. createTemplate

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

3. TemplateCache.tex

describe class TemplateCache.tex here

3.1. contains

Name	`TemplateCache.tex::contains(type:  TClassRef)`
Visibility	public
Parameters	*type: TClassRef*
Return value	**bool** describe return value
behavior	describe beahviour

3.2. get

Name	`TemplateCache.tex::get(type: TClassRef)`
Visibility	public
Parameters	*type: TClassRef*
Return value	**Local**<**FunctionTemplate**> describe return value
behavior	describe beahviour

3.3. store

Name	`TemplateCache.tex::store(type: TClassRef, tpl: Local<FunctionTemplate>)`
Visibility	public
Parameters	*type: TClassRef, tpl: Local<FunctionTemplate>*
Return value	**none**
behavior	describe beahviour

4. ProxyFunctionFactory.tex

describe class ProxyFunctionFactory.tex here

4.1. createProxyFunction

Name	`ProxyFunctionFactory.tex::createProxyFunction(info:  TMethod)`
Visibility	public
Parameters	*info: TMethod*
Return value	**ProxyFunciton** describe return value
behavior	describe beahviour

4.2. fromArgs

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

5. ProxyFunction.tex

describe class ProxyFunction.tex here

5.1. ProxyFunction

Name	`ProxyFunction.tex::ProxyFunction(address: void*, info: TFunction)`
Visibility	public
Parameters	*address: void*, info: TFunction*
Return value	**«constructor»** describe return value
behavior	describe beahviour

5.2. convertArgs

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

5.3. call

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

5.4. isTemplateFunction

Name	`ProxyFunction.tex::isTemplateFunction()`
Visibility	public
Parameters	*none*
Return value	**bool** describe return value
behavior	describe beahviour

6. ProxyObjectFactory.tex

describe class ProxyObjectFactory.tex here

6.1. createProxyObject

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

7. ProxyObject.tex

describe class ProxyObject.tex here

7.1. ProxyObject

Name	`ProxyObject.tex::ProxyObject(address: void*, type: TDataMember)`
Visibility	protected
Parameters	*address: void*, type: TDataMember*
Return value	«**constructor**» describe return value
behavior	describe beahviour

7.2. getAddress

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

7.3. getType

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

7.4. set

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

7.5. get

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

7.6. isPrimitive

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

8. FunctionHelper.tex

describe class FunctionHelper.tex here

8.1. GetCallFunc

Name	`FunctionHelper.tex::GetCallFunc(method: TCppMethod)`
Visibility	public
Parameters	*method: TCppMethod*
Return value	**CallFunc*** describe return value
behavior	describe beahviour

8.2. copyArgs

Name	`FunctionHelper.tex::copyArgs(args: void*, vargs: void**)`
Visibility	public
Parameters	*args: void*, vargs: void***
Return value	**void** describe return value
behavior	describe beahviour

8.3. FastCall

Name	`FunctionHelper.tex::FastCall(method: TCppMethod, args: void*, self: void*, result: void*)`
Visibility	public
Parameters	*method: TCppMethod, args: void*, self: void*, result: void**
Return value	**bool** describe return value
behavior	describe beahviour

8.4. CallV

Name	`FunctionHelper.tex::CallV(method: TCppMethod, self: TCppObject, args: void*)`
Visibility	public
Parameters	*method: TCppMethod, self: TCppObject, args: void**
Return value	**void** describe return value
behavior	describe beahviour

8.5. CallR

Name	`FunctionHelper.tex::CallR(method: TCppMethod, self: TCppObject, args: void*)`
Visibility	public
Parameters	*method: TCppMethod, self: TCppObject, args: void**
Return value	**void*** describe return value
behavior	describe beahviour

8.6. CallS

Name	`FunctionHelper.tex::CallS(method: TCppMethod, self: TCppObject, args: void*)`
Visibility	public
Parameters	*method: TCppMethod, self: TCppObject, args: void**
Return value	**Char*** describe return value
behavior	describe beahviour

8.7. CallO

Name	`FunctionHelper.tex::CallO(method: TCppMethod, self: TCppObject, args: void*, resultype: TCppType)`
Visibility	public
Parameters	*method: TCppMethod, self: TCppObject, args: void*, resultype: TCppType*
Return value	**TCppObject** describe return value
behavior	describe beahviour

8.8. CallConstructor

Name	`FunctionHelper.tex::CallConstructor(method: TCppMethod, klass: TCppType, args: void*)`
Visibility	public
Parameters	*method: TCppMethod, klass: TCppType, args: void**
Return value	**TCppObject** describe return value
behavior	describe beahviour

8.9. CallDestructor

Name	`FunctionHelper.tex::CallDestructor(type: TCppType, self: TCppObject)`
Visibility	public
Parameters	*type: TCppType, self: TCppObject*
Return value	**void** describe return value
behavior	describe beahviour

8.10. IsConstructor

Name	`FunctionHelper.tex::IsConstructor(method:  TCppMethod)`
Visibility	public
Parameters	*method: TCppMethod*
Return value	**bool** describe return value
behavior	describe beahviour

8.11. IsPublicMethod

Name	`FunctionHelper.tex::IsPublicMethod(method: TCppMethod)`
Visibility	public
Parameters	*method: TCppMethod*
Return value	**bool** describe return value
behavior	describe beahviour

8.12. IsStaticMethod

Name	`FunctionHelper.tex::IsStaticMethod(method:  TCppMethod)`
Visibility	public
Parameters	*method: TCppMethod*
Return value	**bool** describe return value
behavior	describe beahviour

8.13. IsConstMethod

Name	`FunctionHelper.tex::IsConstMethod(method:  TCppMethod)`
Visibility	public
Parameters	*method: TCppMethod*
Return value	**bool** describe return value
behavior	describe beahviour

8.14. IsMethodTemplate

Name	`FunctionHelper.tex::IsMethodTemplate(method: TCppMethod)`
Visibility	public
Parameters	*method: TCppMethod*
Return value	**bool** describe return value
behavior	describe beahviour

8.15. GetMethodNumTemplateArgs

Name	`FunctionHelper.tex::GetMethodNumTemplateArgs(scope: TCppScope, imeth: TCppIndex)`
Visibility	public
Parameters	*scope: TCppScope, imeth: TCppIndex*
Return value	**TCppIndex** describe return value
behavior	describe beahviour

8.16. GetMethodTemplateArgName

Name	`FunctionHelper.tex::GetMethodTemplateArgName(scope: TCppScope, imeth: TCppIndex, iarg: TCppIndex)`
Visibility	public
Parameters	*scope: TCppScope, imeth: TCppIndex, iarg: TCppIndex*
Return value	**string** describe return value
behavior	describe beahviour

8.17. GetNumMethods

Name	`FunctionHelper.tex::GetNumMethods(scope: TCppScope)`
Visibility	public
Parameters	*scope: TCppScope*
Return value	**TCppIndex** describe return value
behavior	describe beahviour

8.18. GetMethodIndexAt

Name	`FunctionHelper.tex::GetMethodIndexAt(scope: TCppScope, imeth: TCppIndex)`
Visibility	public
Parameters	*scope: TCppScope, imeth: TCppIndex*
Return value	**TCppIndex** describe return value
behavior	describe beahviour

8.19. GetMethodsFromName

Name	`FunctionHelper.tex::GetMethodsFromName(scope: TCppScope, name: string)`
Visibility	public
Parameters	*scope: TCppScope, name: string*
Return value	**vector<TCppMethod>** describe return value
behavior	describe beahviour

8.20. GetMethod

Name	`FunctionHelper.tex::GetMethod(scope: TCppScope, imeth: TCppIndex)`
Visibility	public
Parameters	*scope: TCppScope, imeth: TCppIndex*
Return value	**TCppMethod** describe return value
behavior	describe beahviour

8.21. GetMethodName

Name	`FunctionHelper.tex::GetMethodName(method: TCppMethod)`
Visibility	public
Parameters	*method: TCppMethod*
Return value	**string** describe return value
behavior	describe beahviour

8.22. GetMethodResultType

Name	`FunctionHelper.tex::GetMethodResultType(method: TCppMethod)`
Visibility	public
Parameters	*method: TCppMethod*
Return value	**string** describe return value
behavior	describe beahviour

8.23. GetMethodNumArgs

Name	`FunctionHelper.tex::GetMethodNumArgs(method: TCppMethod)`
Visibility	public
Parameters	*method: TCppMethod*
Return value	**TCppIndex** describe return value
behavior	describe beahviour

8.24. GetMethodReqArgs

Name	`FunctionHelper.tex::GetMethodReqArgs(method:  TCppMethod)`
Visibility	public
Parameters	*method: TCppMethod*
Return value	**TCppIndex** describe return value
behavior	describe beahviour

8.25. GetMethodArgName

Name	`FunctionHelper.tex::GetMethodArgName(method: TCppMethod, iarg: int)`
Visibility	public
Parameters	*method: TCppMethod, iarg: int*
Return value	**string** describe return value
behavior	describe beahviour

8.26. GetMethodArgType

Name	`FunctionHelper.tex::GetMethodArgType(method: TCppMethod, iarg: int)`
Visibility	public
Parameters	*method: TCppMethod, iarg: int*
Return value	**string** describe return value
behavior	describe beahviour

8.27. GetMethodArgDefault

Name	`FunctionHelper.tex::GetMethodArgDefault(method:  TCppMethod, iarg:  int)`
Visibility	public
Parameters	*method: TCppMethod, iarg: int*
Return value	**string** describe return value
behavior	describe beahviour

8.28. GetMethodSignature

Name	`FunctionHelper.tex::GetMethodSignature(scope: TCppScope, imeth: TCppIndex)`
Visibility	public
Parameters	*scope: TCppScope, imeth: TCppIndex*
Return value	**string** describe return value
behavior	describe beahviour

9. ClassHelper.tex

describe class ClassHelper.tex here

9.1. IsNamespace

Name	`ClassHelper.tex::IsNamespace(scope:  TCppScope)`
Visibility	public
Parameters	*scope: TCppScope*
Return value	**bool** Returns a boolean which checks if the scope represents a namespace.
behavior	Checks if scope represents a namespace.

9.2. IsAbstract

Name	`ClassHelper.tex::IsAbstract(klass: TCppType)`
Visibility	public
Parameters	*klass: TCppType*
Return value	**bool** describe return value
behavior	Checks if klass is an abstract class, and hence can be instantiated or not.

9.3. IsEnum

Name	`ClassHelper.tex::IsEnum(typename: string)`
Visibility	public
Parameters	*typename: string*
Return value	**bool** describe return value
behavior	Checks if string represents a enum.

9.4. IsStruct

Name	`ClassHelper.tex::IsStruct(typename:  string)`
Visibility	public
Parameters	*typename: string*
Return value	**bool** describe return value
behavior	describe beahviour

9.5. GetFinalName

Name	`ClassHelper.tex::GetFinalName(klass: TCppType)`
Visibility	public
Parameters	*klass: TCppType*
Return value	**string** describe return value
behavior	describe beahviour

9.6. GetScopedFinalName

Name	`ClassHelper.tex::GetScopedFinalName(klass:  TCppType)`
Visibility	public
Parameters	*klass: TCppType*
Return value	**string** describe return value
behavior	describe beahviour

9.7. GetNumBases

Name	`ClassHelper.tex::GetNumBases(klass:  TCppType)`
Visibility	public
Parameters	*klass: TCppType*
Return value	**TCppIndex** describe return value
behavior	describe beahviour

9.8. GetBaseName

Name	`ClassHelper.tex::GetBaseName(klass:  TCppType, ibase:  TCppIndex)`
Visibility	public
Parameters	*klass: TCppType, ibase: TCppIndex*
Return value	**string** describe return value
behavior	describe beahviour

9.9. IsSubtype

Name	`ClassHelper.tex::IsSubtype(derived: TCppType, base: TCppType)`
Visibility	public
Parameters	*derived: TCppType, base: TCppType*
Return value	**bool** describe return value
behavior	describe beahviour

9.10. GetNumDatamembers

Name	`ClassHelper.tex::GetNumDatamembers(scope: TCppScope)`
Visibility	public
Parameters	*scope: TCppScope*
Return value	**TCppIndex** describe return value
behavior	describe beahviour

9.11. GetDatamemberName

Name	`ClassHelper.tex::GetDatamemberName(scope: TCppScope, idata: TCppIndex)`
Visibility	public
Parameters	*scope: TCppScope, idata: TCppIndex*
Return value	**string** describe return value
behavior	describe beahviour

9.12. GetDatamemberType

Name	`ClassHelper.tex::GetDatamemberType(scope: TCppScope, idata: TCppIndex)`
Visibility	public
Parameters	*scope: TCppScope, idata: TCppIndex*
Return value	**string** describe return value
behavior	describe beahviour

9.13. GetDatamemberOffset

Name	`ClassHelper.tex::GetDatamemberOffset(scope: TCppScope, idata: TCppIndex)`
Visibility	public
Parameters	*scope: TCppScope, idata: TCppIndex*
Return value	**ptrdiff** describe return value
behavior	describe beahviour

9.14. GetDatamemberIndex

Name	`ClassHelper.tex::GetDatamemberIndex(scope: TCppScope, name: string)`
Visibility	public
Parameters	*scope: TCppScope, name: string*
Return value	**TCppIndex** describe return value
behavior	describe beahviour

9.15. IsPublicData

Name	`ClassHelper.tex::IsPublicData(scope: TCppScope, idata: TCppIndex)`
Visibility	public
Parameters	*scope: TCppScope, idata: TCppIndex*
Return value	**bool** describe return value
behavior	describe beahviour

9.16. IsStaticData

Name	`ClassHelper.tex::IsStaticData(scope: TCppScope, idata: TCppIndex)`
Visibility	public
Parameters	*scope: TCppScope, idata: TCppIndex*
Return value	**bool** describe return value
behavior	describe beahviour

9.17. IsConstData

Name	`ClassHelper.tex::IsConstData(scope: TCppScope, idata: TCppIndex)`
Visibility	public
Parameters	*scope: TCppScope, idata: TCppIndex*
Return value	**bool** describe return value
behavior	describe beahviour

9.18. IsEnumData

Name	`ClassHelper.tex::IsEnumData(scope: TCppScope, idata: TCppIndex)`
Visibility	public
Parameters	*scope: TCppScope, idata: TCppIndex*
Return value	**bool** describe return value
behavior	describe beahviour

9.19. resolveAddress

Name	`ClassHelper.tex::resolveAddress(staticMember: TDataMember, clazz: TClassRef)`
Visibility	public
Parameters	*staticMember: TDataMember, clazz: TClassRef*
Return value	**void*** describe return value
behavior	describe beahviour

9.20. resolveAddress

Name	`ClassHelper.tex::resolveAddress(staticMember: TDataMember, clazz: TClassRef)`
Visibility	public
Parameters	*staticMember: TDataMember, clazz: TClassRef*
Return value	**void*** describe return value
behavior	describe beahviour