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. ProxyObjectFactory

The ProxyObjectFactory is used whenever a v8 objects needs to be converted into it's ROOT counterpart or vice versa. In a lot of cases we have a memory addresses and types, described by a string. The ProxyObjectFactory needs to parse the type string in order to decide on a class (these classes shall be called ProxyObjects) to which the void type variable needs to be forwarded. These ProxyObjects cotain the correct semantics to generate ROOT and Node objects. They are all named [type]ProxyObject derive from the class ProxyObject. The following ProxyObjects need to implemented to make the bindings work:

- **String**ProxyObject
- **Number**ProxyObject
- **Bool**ProxyObject
- **Object**ProxyObject
- ...

1.1. createProxyObject

Name	`ProxyObjectFactory::createProxyObject(void* obj, std::string type))`
Visibility	Public static
Parameters	**obj**: The object to be converted **type**: The typename, provided by ROOT
Return value	**ProxyObject** The ProxyObjects that matches the given type. The ProxyObject has already been initialized with the obj pointer.
behavior	Decides which ProxyObject can handle the obj pointer and returns an instance of this

Before creating a new ProxyObject this should querry the ProxyObjectCache, checking if the object at the given momeory address has already been proxied.

2. ProxyObject

ProxyObject is an interace defining the following abstract methods:

2.1. isScalar

Name	`ProxyObject::isScalar())`
Visibility	Public abstract
Parameters	*none*
Return value	**bool** true: The object is scalar, no recursion is needed to create a ROOT/v8 representation
behavior	This is usually just a return statement, as String, Number, Bool, ... ProxyObjects will always handle scalar data and ObjectProxyObjets will be the only non scalar ProxyObjects (and will therefor return false)

2.2. getV8Handle

Name	`ProxyObject::getV8Handle())`
Visibility	Public abstract
Parameters	*none*
Return value	**v8::Handle** A Handle will be generated continaing the data, used to initialize the ProxyObject.
behavior	This highly depends on the obecjt's type scalar types might just call a constructor and return the result, ObjectProxyObjects will need to step down all through the objects children.

3. Appendix

3.1. Glossary