The Grantlee Template System

Stephen Kelly

What can Grantlee do?

How will applications use it?

How does it work?

Demo

The old way

```
QString str;
str.append(“<html><h1>” + title + “</h1>”);
str.append(“<div>” + content + “</div>”);
str.append(“</html>”);
```

The old way

```
QString str;
str.append(“<html><h1>” + title + “</h1>”);
str.append(“<div>” + content + “</div>”);
str.append(“</html>”);
```

Goals

- Simplify theming for application developers
- Separate exported data from its presentation
- Use syntax of the Django Template system

The Grantlee Template System

Artist

Developer

Application Code

```
Engine *engine = Engine::instance();
Template *t = engine->getTemplate(“template.html”)
contextMap.insert(“entities”, selectedEntities);
Context c(contextMap);
return t->render(&c);
```

Template Syntax

```
<html>
{# A simple comment #}
<div>{% for entity in entities %}
  {% if entity.isBook %}
   <h1>{{ entity.title|upper }}</h1>
    {% include “booktemplate.html” %}
  {% else %}
     {% include “pagetemplate.html” %}
  {% endif %}
{% endfor %}</div>
</html>
```

Tokens

- Text: “<html>”, etc
- Comment: “{# A simple comment #}”
- Variable (with filter): “{{ entity.title|upper }}”
- Tag: “{% for entity in entities %}”,
  “{% if entity.isBook %}”, etc

The Grantlee Template System

Python

Web framework

C++

Rich client framework

The Grantlee Template System

Python

Web framework

Dynamic typing

C++

Rich client framework

Static typing

The Grantlee Template System

Python

Web framework

Dynamic typing

Runtime Introspection

Everything is an Object

C++

Rich client framework

Static typing

Compiled binaries

Similar interfaces for similar objects

Introspection

{{ var }}	object.__str__()	QVariant.toString()
{{ map.key }}	dictionary	QVariantHash
{{ list.2 }}	list	QVariantList
{{ obj.prop }}	getattr	QObject* property

Introspection

```
class KJotsEntity : public QObject
{
  Q_OBJECT
  Q_PROPERTY( QString title READ title )
  Q_PROPERTY( bool isBook READ isBook )
  Q_PROPERTY( bool isPage READ isPage )
  Q_PROPERTY( QString content READ content )
  Q_PROPERTY( QVariantList entities READ entities )

public:
  bool isBook() const;
  bool isPage() const;
  QString title() const;
  QString content() const;
  QVariantList entities() const;
};
```

Introspection

```
class KJotsEntity : public QObject
{
  Q_OBJECT
  Q_PROPERTY( QString title READ title )
  Q_PROPERTY( bool isBook READ isBook )
  Q_PROPERTY( bool isPage READ isPage )
  Q_PROPERTY( QString content READ content )
  Q_PROPERTY( QVariantList entities READ entities )

public:
  bool isBook() const;
  bool isPage() const;
  QString title() const;
  QString content() const;
  QVariantList entities() const;
};
```

Custom Tags and Filters

Import modules

register.tag(...)

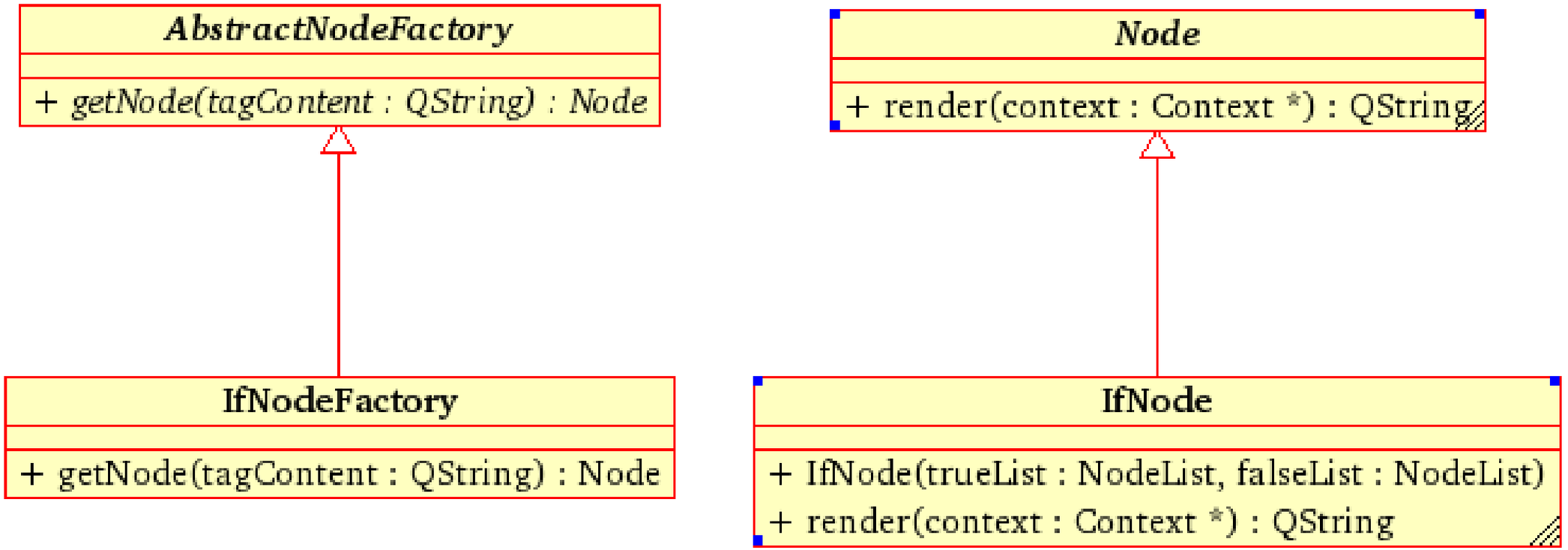
Python

QtPlugin

Class interface

C++ / QtScript

Custom Tags and Filters



Custom Tags and Filters

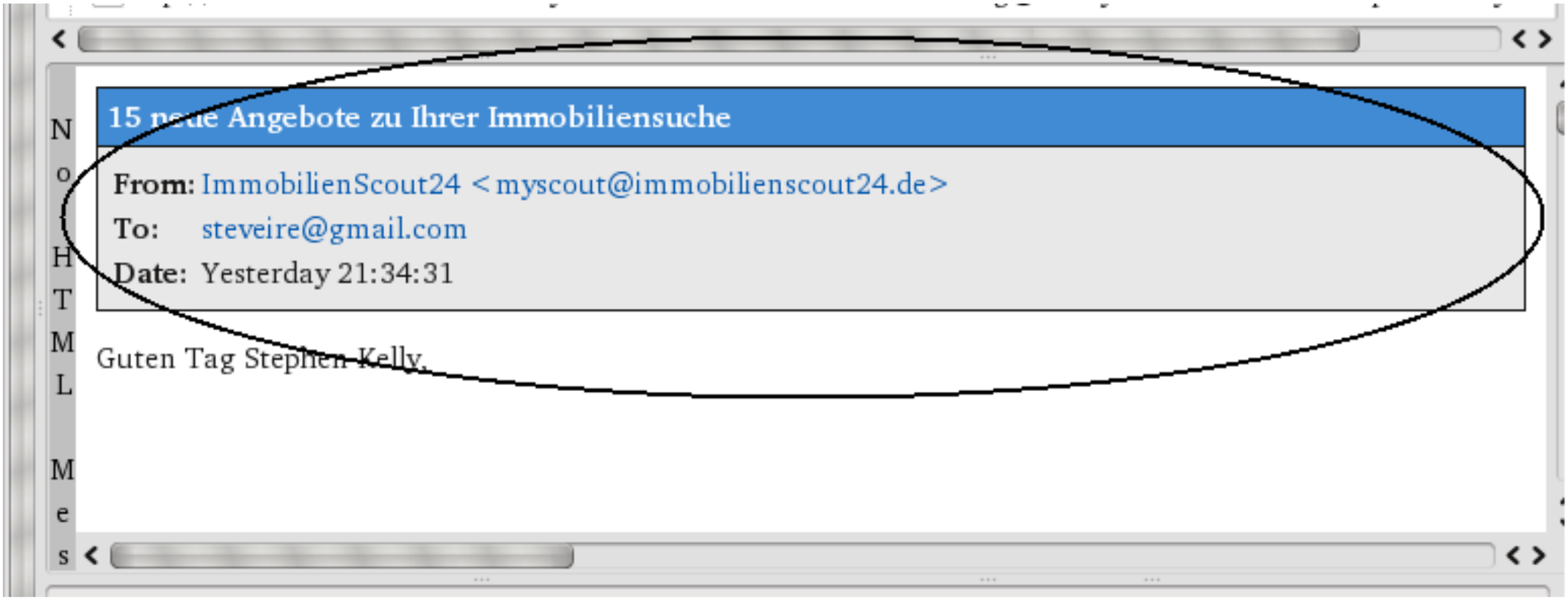
```
#include <grantlee/interfaces/taglibraryinterface.h>

class MyLibrary : public TagLibraryInterface
{
  Q_OBJECT
  Q_INTERFACES( Grantlee::TagLibraryInterface )
public:
  MyLibrary();

  QHash<QString, AbstractNodeFactory*> nodeFactories();

  QHash<QString, Filter*> filters();
};
```

Custom Tags and Filters

```
#include <grantlee/interfaces/taglibraryinterface.h>

class MyLibrary : public TagLibraryInterface
{
  Q_OBJECT
  Q_INTERFACES( Grantlee::TagLibraryInterface )
public:
  MyLibrary();

  QHash<QString, AbstractNodeFactory*> nodeFactories();

  QHash<QString, Filter*> filters();
};
```

Consumers

- Kjots, KMail, KNode, Akregator, etc



Consumers

- Kjots, KMail, KNode, Akregator, etc
- Complex MIME message structures

```
To: {{ recipient }}
Date: {% now %}
Subject: {{ subject }}
{% for header in extraHeaders %}
{{ header.key }}: {{header.value }}
{% endfor %}

{% for part in message %}
  {% include part.typeTemplate %}
{% endfor %}
```

Consumers

- Kjots, KMail, KNode, Akregator, etc
- Complex MIME message structures
- Akonadi mail-merge agent

```
Dear {{ customer.contactName }},
We are out of stock of {{ product }}, so your orders on
{% for order in customer.affectedOrders %}
 * {{ order.date }}
{% endfor %}
Can not be filled.
```

Summary

- Successful port of corelib and default tags
- Extensible with QtPlugin and QtScript
- Covered by LGPLv3
- 350 unit tests ported
- Performance tests with QBENCHMARK
- Git repo available (**http://www.grantlee.org**)
- KMail and KJots in KDE 4.4
- Version 0.1 release 'soon'

Questions

&

Answers