Cookbook - adding language

- Enabling new language in Storefront
  - Locale specific property files
  - Configure language specific email templates
- Resources

Enabling new language in Storefront

Storefront and REST API are using Java SDK resource bundles to support localisations. Therefore the task of enabling new interface language requires:

- Provision of locale specific property files
- Configuration of LanguageService Spring bean
- *Optionally* Enabling language for shop in Admin App
- *Optionally* Configure language specific email templates

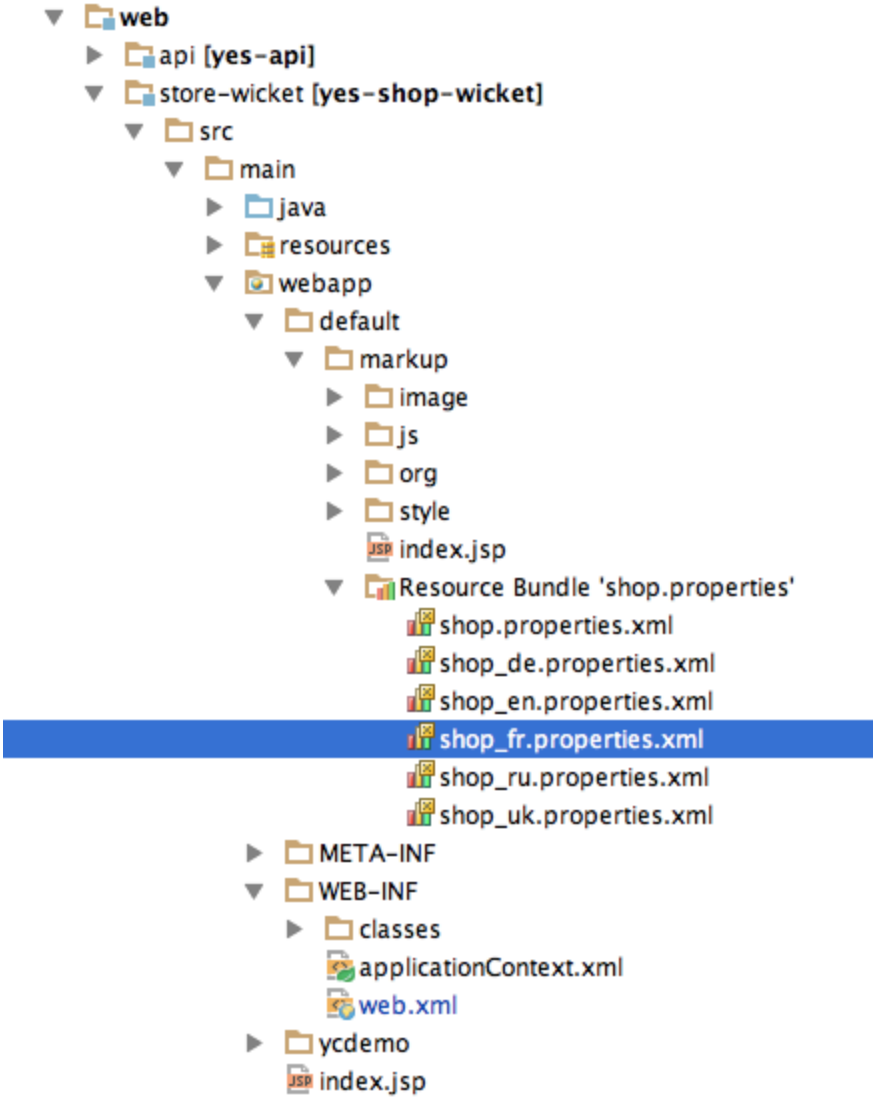
Locale specific property files

All property files for interface are located in theme specific directories:

```
YC_HOME/web/store-wicket/src/main/webapp/default/markup/
```

There is only a single file for each language which is named shop_[language].properties.xml. For fr language this would be shop_fr.properties.xml

At the very least this file has to be present in default theme. If there are theme specific translations then those themes should also define such file.



The contents of this file is in XML format but follow the same principle of key value pairs:

Example shop_en.properties.xml

```
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
    <!-- Core wicket -->
    <entry key="Required">'${label}' is required</entry>
    <entry key="IConverter">'${input}' is not a valid ${type}</entry>
    <entry key="RangeValidator">${input} is not between ${minimum} and
${maximum}</entry>
    <entry key="MinimumValidator">'${input}' is smaller than the minimum of
${minimum}</entry>
    <entry key="MaximumValidator">'${input}' is larger than the maximum of
${maximum}</entry>
    <entry key="NumberValidator.positive">'${input}' must be
positive</entry>
    <entry key="NumberValidator.negative">'${input}' must be
negative</entry>
    <entry key="StringValidator.range">'${input}' is not between ${minimum}
and ${maximum} characters long</entry>
    <entry key="StringValidator.minimum">'${input}' is shorter than the
minimum of ${minimum} characters</entry>
    <entry key="StringValidator.maximum">'${input}' is longer than the
maximum of ${maximum} characters</entry>
    <entry key="StringValidator.exact">'${input}' is not exactly ${exact}
characters long</entry>
    <entry key="DateValidator.range">'${input}' is not between ${minimum}
and ${maximum}</entry>
    <entry key="DateValidator.minimum">'${input}' is less than the minimum
of ${minimum}</entry>
...
</properties>
```

Configuration of LanguageService

> As of 3.2.0+ Below configuration have been moved to maven environment specific filters, namely: yc-config.properties

When translations resource bundle is available we can reference it in the LanguageService Spring configuration located in

```
YC_HOME/core/src/main/resources/core-services.xml
```

This configuration allows to define language name for storefront language widget and configure default language configurations

```
```

```
...
    <bean id="languageService"
class="org.yes.cart.service.misc.impl.LanguageServiceImpl">
        <constructor-arg index="0">
            <map>
                <entry key="en" value="Eng"/>
                <entry key="de" value="Deu"/>
                <entry key="ru" value="&#x0420;&#x0443;&#x0441;"/>
                <entry key="uk" value="&#x0423;&#x043A;&#x0440;"/>
<!--
   1. Language name for language widget
  -->
                <entry key="fr" value="Fra"/>
            </map>
        </constructor-arg>
        <constructor-arg index="1">
            <map>
                <entry key="DEFAULT">
                    <list>
                        <!-- by convention the 0th element is the default
language -->
                        <value>en</value>
                        <value>de</value>
                        <value>ru</value>
                        <value>uk</value>
<!--
   2. Declare that French language is included by default
  -->
                        <value>fr</value>
                    </list>
                </entry>
                <entry key="SHOP10">
                    <list>
                        <!-- by convention the 0th element is the default
language -->
                        <value>en</value>
                        <value>de</value>
                        <value>ru</value>
                        <value>uk</value>
<!--
   3. Optionally declare that French language is included by default for
specific shop
  -->
                        <value>fr</value>
                    </list>
                </entry>
            </map>
        </constructor-arg>
        <constructor-arg index="2" ref="shopService"/>
    </bean>
...
```

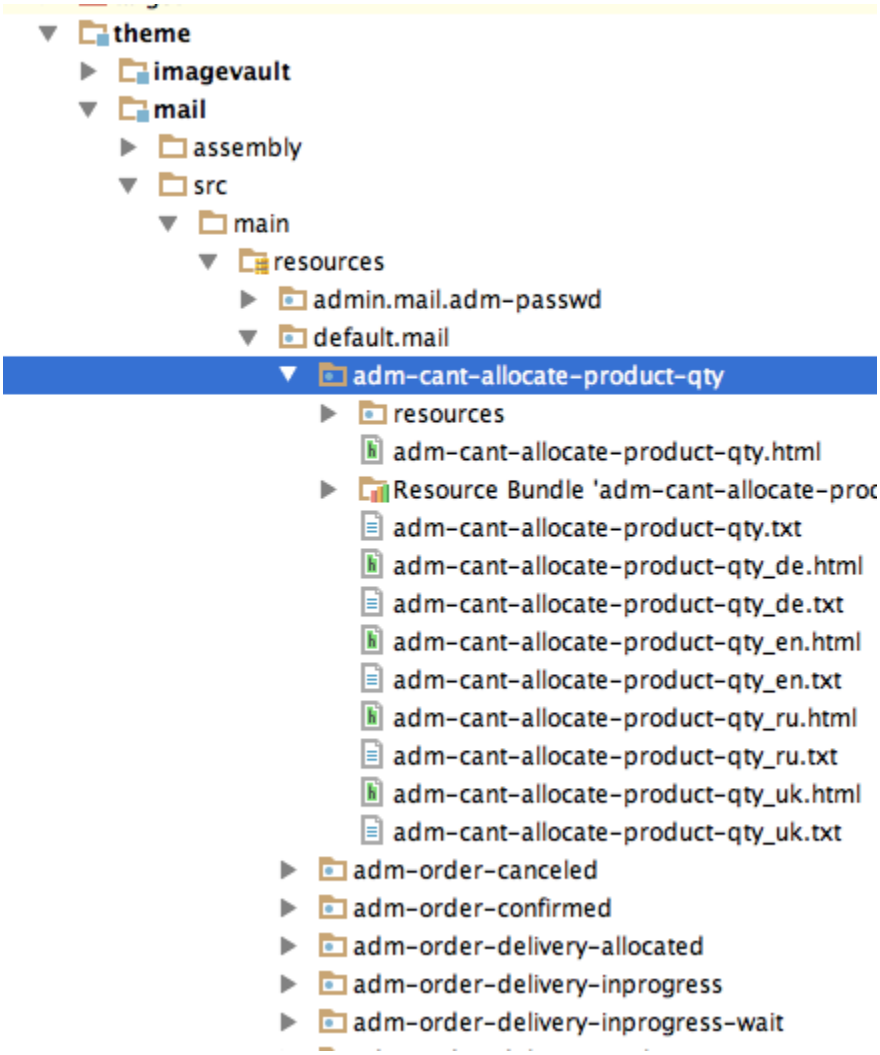
Enabling language for shops

With above configurations French will become available to shops by default. To suppress use of this language it is possible to specify explicitly which languages are available in shop configurations in Admin App.

Configure language specific email templates

Recommended approach for email templates configuration is via CMS. For more details on email see theme documentation.

If however you wish to provide a complete default language specific theme you need to create language specific email templates in "theme/mail" module.



Each email bundle is contained within a directory. Language specific templates are specified by appending language suffix.

So for French (fr) template for "adm-cant-allocate-product-qty" email we would need to specify:

- adm-cant-allocate-product-qty_fr.html

- adm-cant-allocate-product-qty_fr.txt
- adm-cant-allocate-product-qty_fr.properties

Resources

- Admin App manual
- Themes manual
- Installation from source guide maven section