TO THE NEW™ DIGITAL

Migrating Plugins to Grails 3

Giving Back to Grails Community

- by Migrating Plugins to Grails 3

About Me

Puneet Behl

Associate Technical Lead

TO THE NEW Digital

puneet.behl@tothenew.com

GitHub: https://github.com/puneetbehl/

Twitter: @puneetbhl

LinkedIn: https://in.linkedin.com/in/puneetbhl

Agenda

- Background
- Important things
- Let’s migrate
- Publishing it
- Few useful tips

How it all started?

- Few folks attended GR8Conf last year
- Grails 3 launched
- git clone Grails :-)
- Could not migrate :(- because old plugins not supported
- Someone need to migrate == opportunity
- Migration == read documentation :-)

What so far??

- Migrated around 8+ plugins to Grails 3, including Jodatime, RabbitMQ, Elasticsearch, Grails Asynchronous Mail, Export, Grails Remote Control, CKEditor etc.
- Organized a whole day workshop, where we migrated around 6-7 plugins in a day
- Leading a team of enthusiasts involved in migrating 21+ plugins to Grails 3

Things to consider

- Removal of before & after interceptors
- Project structure
- File locations
- Configurations
- Package name
- Legacy Gant Scripts
- Gradle Build Systems

File Location Differences

Grails 2	Grails 3
grails-app/conf/BuildConfig.groovy	build.gradle
grails-app/conf/Config.groovy	grails-app/config/application.groovy
grails/conf/UrlMappings.groovy	grails-app/controllers/UrlMappings.groovy
grails-app/conf/BootStrap.groovy	grails-app/init/BootStrap.groovy
scripts	src/main/scripts
src/groovy	src/main/groovy
src/java	src/main/java

File Location Differences

Grails 2	Grails 3
test/unit	src/test/groovy
test/integration	src/integration-test/groovy
web-app	src/main/webapp or src/main/resources/
*GrailsPlugin.groovy	src/main/groovy/

New Files in Grails 3

File	Description
build.gradle	The Gradle build descriptor location in the root of project. Replaces the ***BuildConfig.groovy***
gradle.properties	Properties file defining the Grails and Gradle versions
grails-app/conf/logback.groovy	Logging previously defined in Config.groovy is now defined using Logback, replacing log4j
grails-app/conf/application.(yml, groovy)	Configuration can now also be defined using YAML or Groovy
grails-app/init/<package>/Application.groovy	The class used By Spring Boot to start the application

Files Not Present in Grails 3

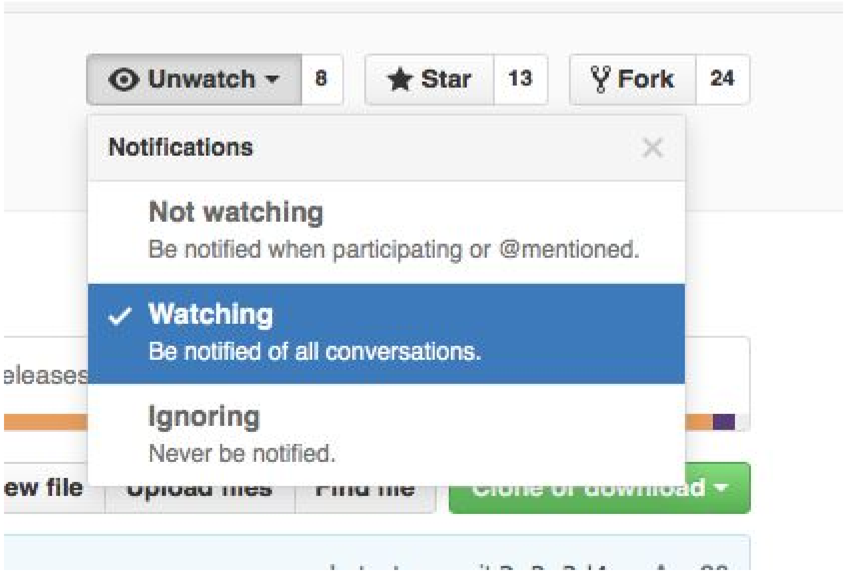
File	Description
application.properties	The application name and version is now defined in build.gradle
grails-app/conf/DataSource.groovy	Merged together into application.yml or application.groovy
lib/	Dependency resolution should be used to resolve JAR files
web-app/WEB-INF/applicationContext.xml	Removed, beans can be defined in grails-app/conf/spring/resources.groovy

Files Not Present in Grails 3

File	Description
src/templates/war/web.xml	Grails 3.0 no longer requires web.xml. Customizations can be done via Spring
web-app/WEB-INF/sitemesh.xml	Removed, sitemesh filter no longer present.
web-app/WEB-INF/tld	Removed, can be restored in src/main/webapp or src/main/resources/WEB-INF

Few things, I prefer to do before migration

- **Fork the existing repository**
- **Star it** - Bookmark it so it's easy to find
- **Watching** - be notified of all the conversations
- Create an issue in existing plugin, "Migrate to Grails 3" and mention that you are working on it



Notifications

Not watching
Be notified when participating or @mentioned.

✓ Watching
Be notified of all conversations.

Ignoring
Never be notified.

Create a new branch for previous version on forked repo

- Note the previous version of plugin from the plugin descriptor file
- Create a new branch for migration. Run the following commands if you’ve already forked and cloned the repo on your dev machine

```
git pull origin master
git checkout -b 1.0.x
git push origin 1.0.x
git checkout master
```

How can I create inline plugin in Grails 3?

Inline Plugin...

- In Grails 2.x it was possible to specify inline plugins in BuildConfig

Inline Plugin...

- In Grails 2.x it was possible to specify inline plugins in BuildConfig
- In Grails 3.x inline plugin functionality has been replaced by Gradle's multi-project build feature.

Inline Plugin - In 4 simple steps

- `grails create-app myApp`
- `grails create-plugin myPlugin`
- `vim settings.gradle` & type `include myPlugin`
- Goto `build.gradle` in myApp and add a dependency

  `compile project(:myPlugin)`

Let’s get started

- Create a new plugin in Grails 3 with the same name

```
“grails create-plugin asynchronous-mail”
```

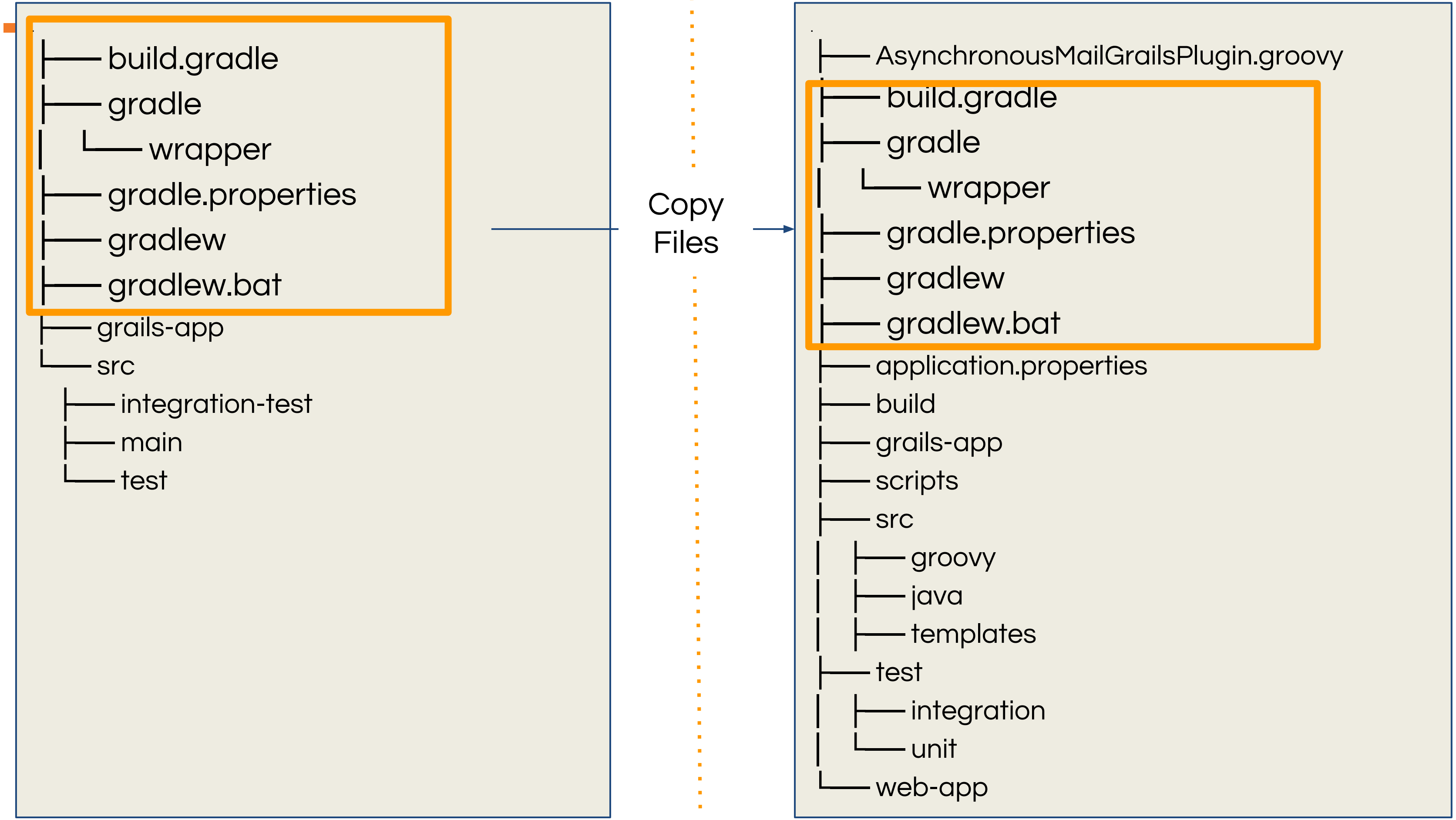
Copy New Files from Grails 3 Backbone project to Forked version...

Barebone Grails 3 Plugin

```
├── build.gradle
├── gradle
│   └── wrapper
├── gradle.properties
├── gradlew
├── gradlew.bat
├── grails-app
│   ├── assets
│   ├── conf
│   ├── controllers
│   ├── domain
│   ├── i18n
│   ├── init
│   ├── services
│   ├── taglib
│   ├── utils
│   └── views
└── src
    ├── integration-test
    ├── main
    └── test
```

Forked Project

```
├── AsynchronousMailGrailsPlugin.groovy
├── application.properties
├── grails-app
│   ├── assets
│   ├── conf
│   ├── domain
│   ├── jobs
│   ├── services
│   └── views
├── scripts
├── src
│   ├── groovy
│   ├── java
│   └── webapp
├── test
│   ├── integration
│   └── unit
└── web-app
    └── css
```



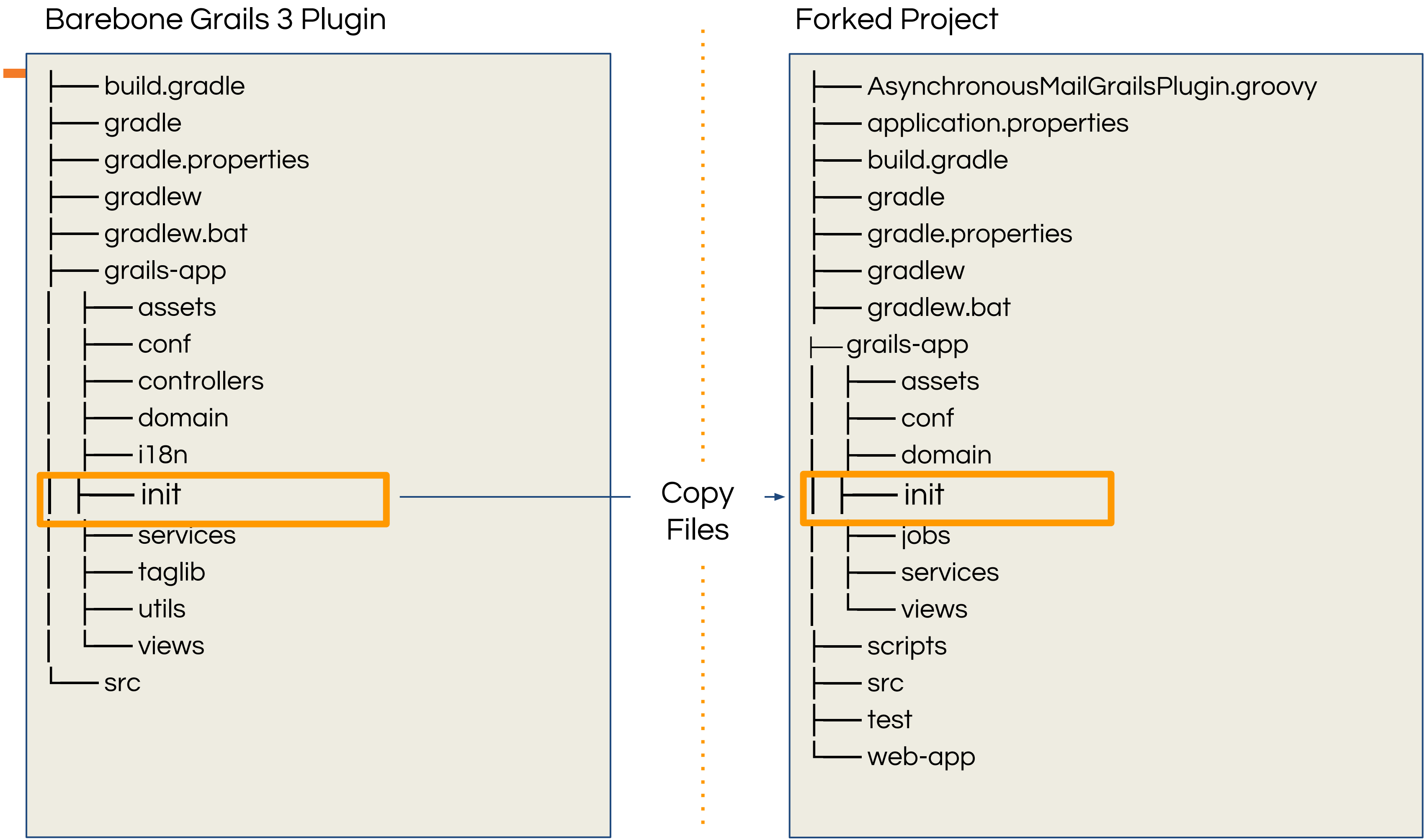
Barebone Grails 3 Plugin

```
.
├── build.gradle
├── gradle
│   └── wrapper
├── gradle.properties
├── gradlew
├── gradlew.bat
├── grails-app
└── src
    ├── integration-test
    ├── main
    └── test
```

Copy Files

Forked Project

```
.
├── AsynchronousMailGrailsPlugin.groovy
├── build.gradle
├── gradle
│   └── wrapper
├── gradle.properties
├── gradlew
├── gradlew.bat
├── application.properties
├── build
├── grails-app
├── scripts
├── src
│   ├── groovy
│   ├── java
│   ├── templates
├── test
│   ├── integration
│   └── unit
└── web-app
```



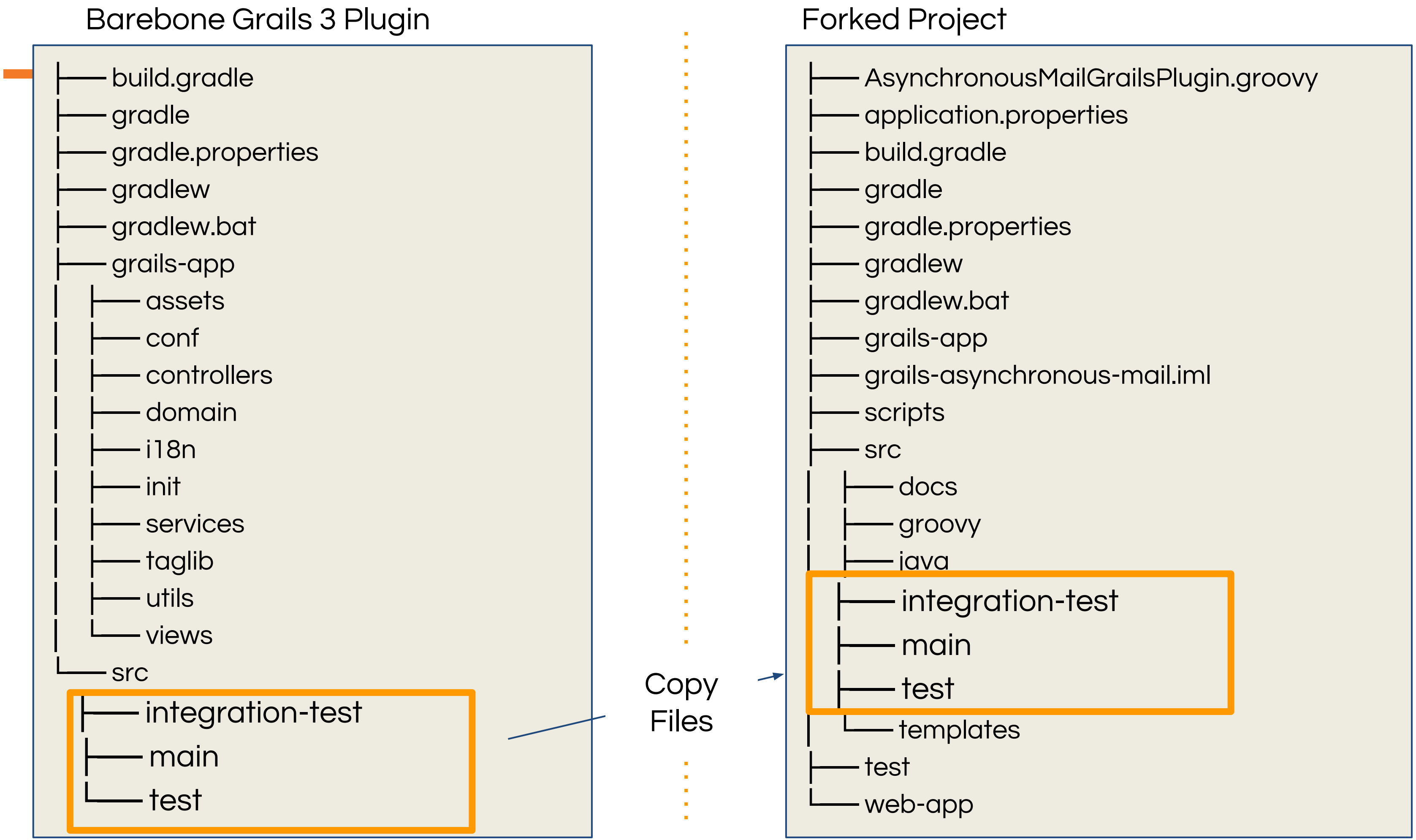
Barebone Grails 3 Plugin

```
├── build.gradle
├── gradle
├── gradle.properties
├── gradlew
├── gradlew.bat
├── grails-app
│   ├── assets
│   ├── conf
│   ├── controllers
│   ├── domain
│   ├── i18n
│   ├── init
│   ├── services
│   ├── taglib
│   ├── utils
│   └── views
└── src
```

Copy Files

Forked Project

```
├── AsynchronousMailGrailsPlugin.groovy
├── application.properties
├── build.gradle
├── gradle
├── gradle.properties
├── gradlew
├── gradlew.bat
├─ grails-app
│   ├── assets
│   ├── conf
│   ├── domain
│   ├── init
│   ├── jobs
│   ├── services
│   └── views
├── scripts
├── src
├── test
└── web-app
```



Barebone Grails 3 Plugin

```
├── build.gradle
├── gradle
├── gradle.properties
├── gradlew
├── gradlew.bat
├── grails-app
│   ├── assets
│   ├── conf
│   ├── controllers
│   ├── domain
│   ├── i18n
│   ├── init
│   ├── services
│   ├── taglib
│   ├── utils
│   └── views
└── src
    ├── integration-test
    ├── main
    └── test
```

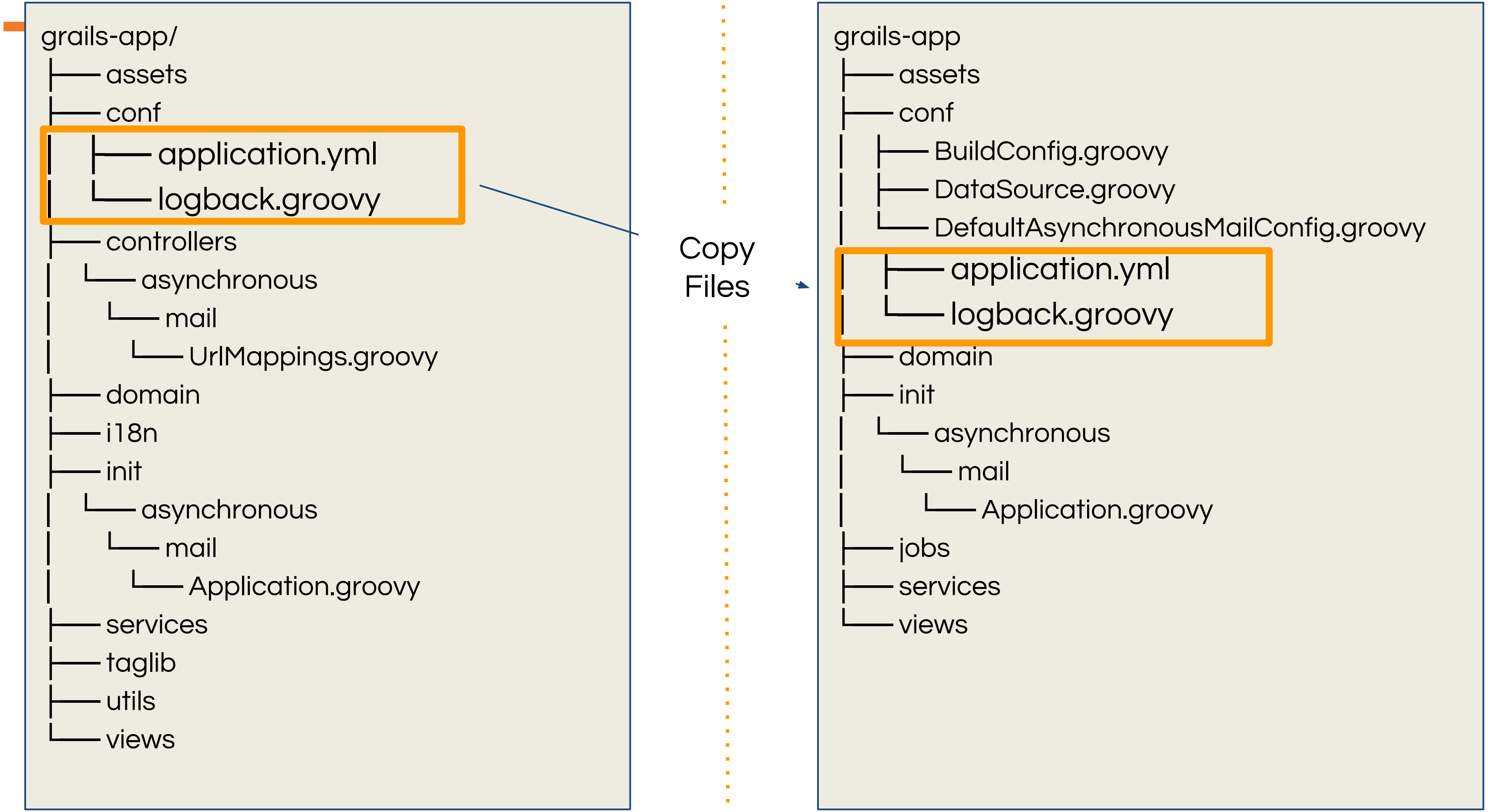
Copy Files

Forked Project

```
├── AsynchronousMailGrailsPlugin.groovy
├── application.properties
├── build.gradle
├── gradle
├── gradle.properties
├── gradlew
├── gradlew.bat
├── grails-app
├── grails-asynchronous-mail.iml
├── scripts
├── src
│   ├── docs
│   ├── groovy
│   ├── java
│   ├── integration-test
│   ├── main
│   ├── test
│   └── templates
├── test
└── web-app
```

Barebone Grails 3 Plugin - grails-app

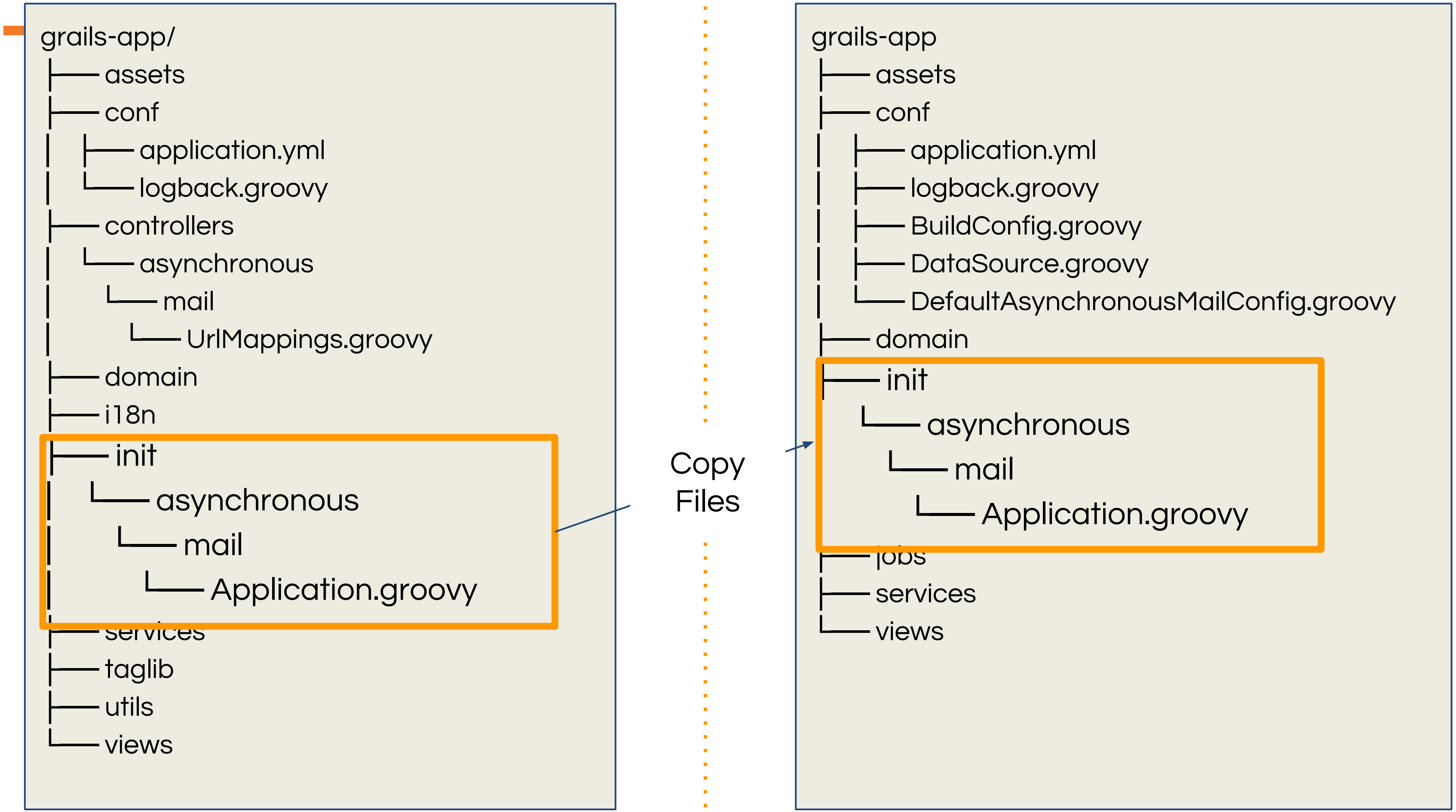
Forked Project - grails-app



```
grails-app/
├── assets
├── conf
│   ├── application.yml
│   └── logback.groovy
├── controllers
│   └── asynchronous
│       └── mail
│           └── UrlMappings.groovy
├── domain
├── i18n
├── init
│   └── asynchronous
│       └── mail
│           └── Application.groovy
├── services
├── taglib
├── utils
└── views
```

Copy Files

```
grails-app
├── assets
├── conf
│   ├── BuildConfig.groovy
│   ├── DataSource.groovy
│   └── DefaultAsynchronousMailConfig.groovy
│   ├── application.yml
│   └── logback.groovy
├── domain
├── init
│   └── asynchronous
│       └── mail
│           └── Application.groovy
├── jobs
├── services
└── views
```



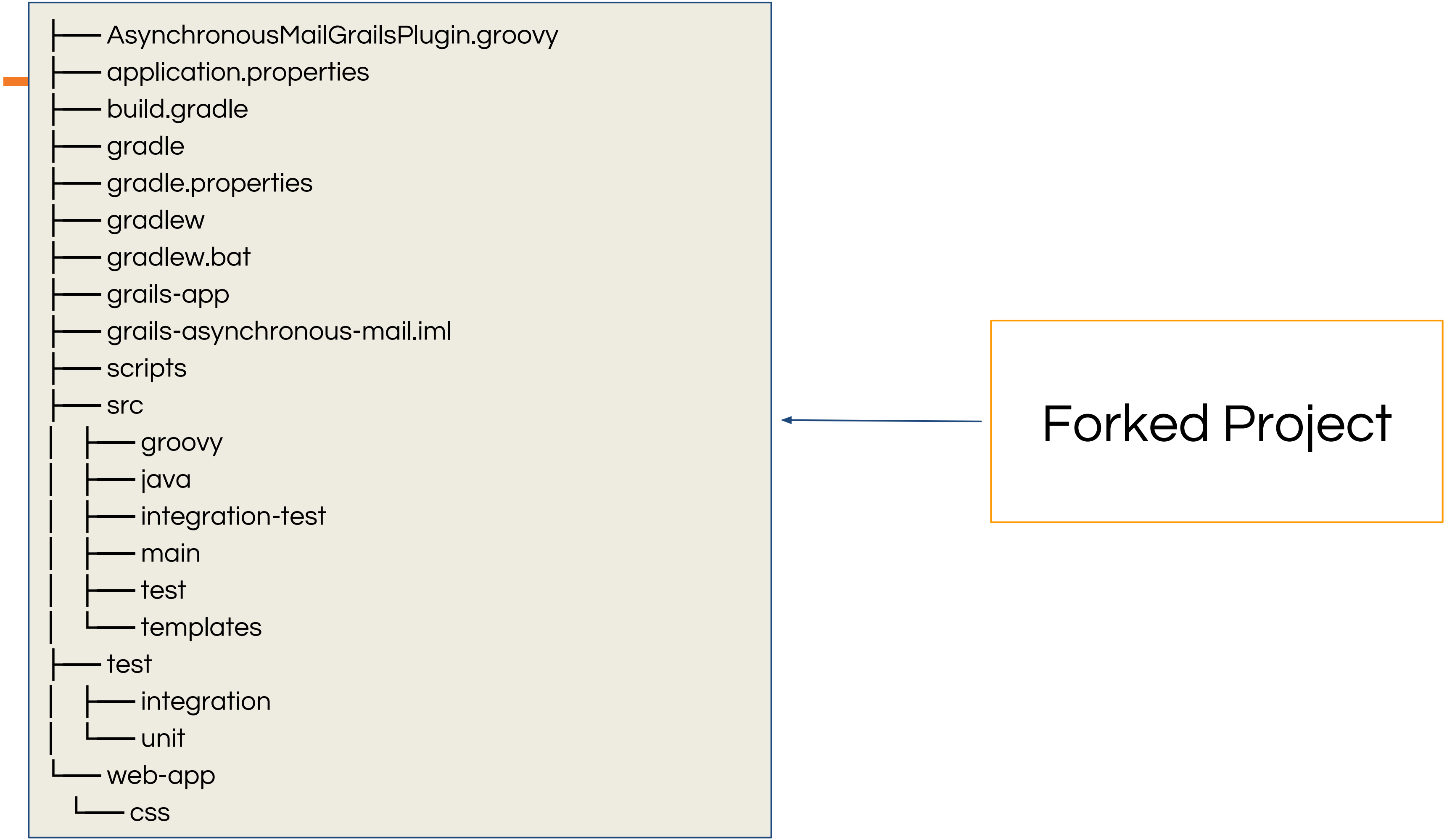
Barebone Grails 3 Plugin

```
grails-app/
├── assets
├── conf
│   ├── application.yml
│   └── logback.groovy
├── controllers
│   └── asynchronous
│       └── mail
│           └── UrlMappings.groovy
├── domain
├── i18n
├── init
│   └── asynchronous
│       └── mail
│           └── Application.groovy
├── services
├── taglib
├── utils
└── views
```

Copy Files

Forked Project

```
grails-app
├── assets
├── conf
│   ├── application.yml
│   ├── logback.groovy
│   ├── BuildConfig.groovy
│   ├── DataSource.groovy
│   └── DefaultAsynchronousMailConfig.groovy
├── domain
├── init
│   └── asynchronous
│       └── mail
│           └── Application.groovy
├── jobs
├── services
└── views
```



```
├── AsynchronousMailGrailsPlugin.groovy
├── application.properties
├── build.gradle
├── gradle
├── gradle.properties
├── gradlew
├── gradlew.bat
├── grails-app
├── grails-asynchronous-mail.iml
├── scripts
├── src
│   ├── groovy
│   ├── java
│   ├── integration-test
│   ├── main
│   ├── test
│   └── templates
├── test
│   ├── integration
│   └── unit
└── web-app
    └── css
```

Forked Project

Now, move sources to new locations...

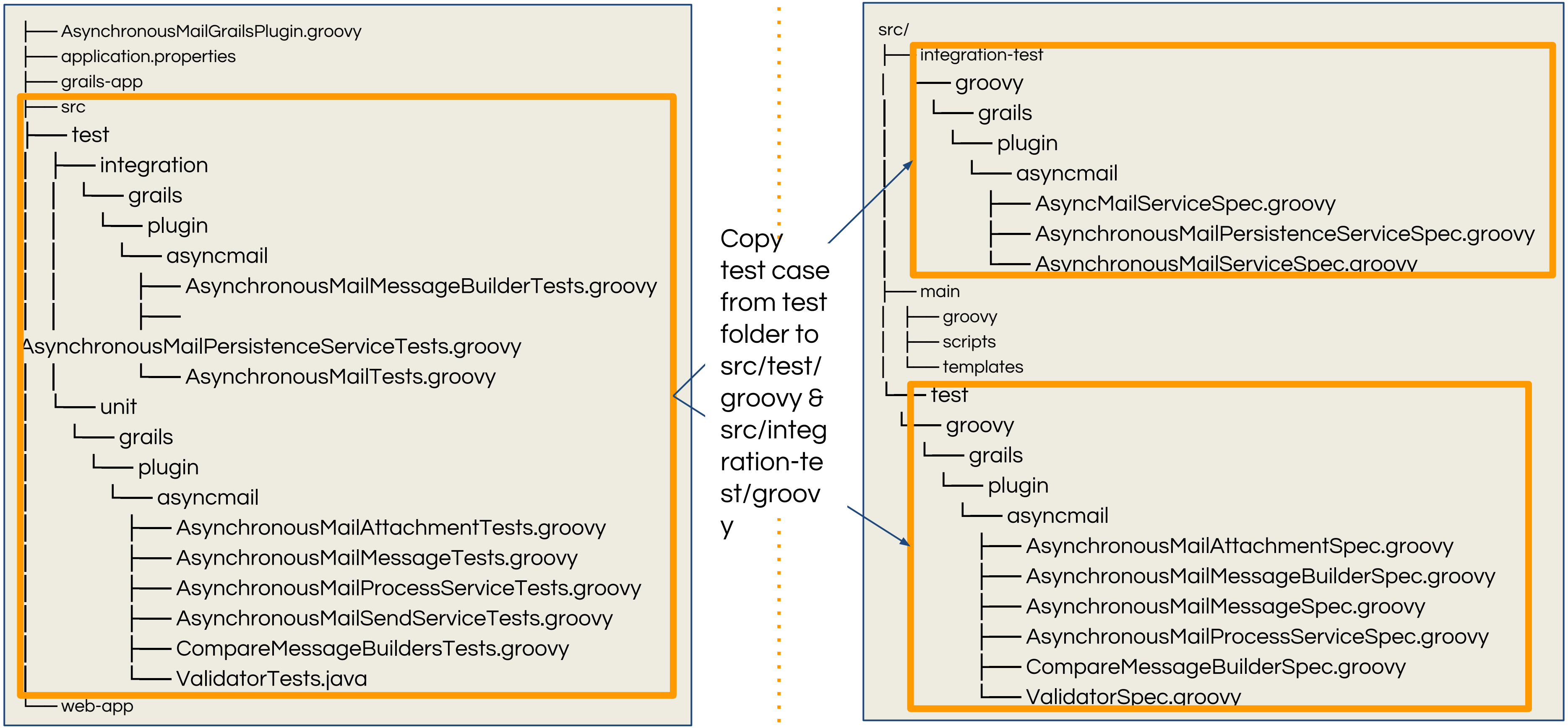
Now, move files from src to src/main

```
src/
├── groovy
│   └── grails
│       └── plugin
│           └── asyncmail
│               ├──
AsynchronousMailMessageBuilder.groovy
│               └──
AsynchronousMailMessageBuilderFactory.groovy
├── java
│   └── grails
│       └── plugin
│           └── asyncmail
│               └── Validator.java
├── templates
└── webapp
```

Copy source files from src/groovy & src/java to src/main/groovy →

```
src/
├── integration-test
│   └── groovy
├── main
│   ├── groovy
│   │   └── grails
│   │       └── plugin
│   │           └── asyncmail
│   │               ├── AsynchronousMailGrailsPlugin.groovy
│   │               ├── AsynchronousMailMessageBuilder.groovy
│   │               ├── AsynchronousMailMessageBuilderFactory.groovy
│   │               └── enums
│   │               │   └── MessageStatus.groovy
│     │              └── Validator.java
│   ├── scripts
│   │   └── CreateAsynchronousMailController.groovy
│   └── templates
│
└── test
    └── groovy
```

Move Unit & Integration Tests



```
├── AsynchronousMailGrailsPlugin.groovy
├── application.properties
├── grails-app
├── src
├── test
│   ├── integration
│   │   └── grails
│   │       └── plugin
│   │           └── asyncmail
│   │               ├── AsynchronousMailMessageBuilderTests.groovy
│   │               ├──
AsynchronousMailPersistenceServiceTests.groovy
│   │               └── AsynchronousMailTests.groovy
│   └── unit
│       └── grails
│           └── plugin
│               └── asyncmail
│                   ├── AsynchronousMailAttachmentTests.groovy
│                   ├── AsynchronousMailMessageTests.groovy
│                   ├── AsynchronousMailProcessServiceTests.groovy
│                   ├── AsynchronousMailSendServiceTests.groovy
│                   ├── CompareMessageBuildersTests.groovy
│                   └── ValidatorTests.java
└── web-app
```

Copy test case from test folder to src/test/groovy & src/integration-test/groovy

```
src/
├── integration-test
│   ── groovy
│   └── grails
│       └── plugin
│           └── asyncmail
│               ├── AsyncMailServiceSpec.groovy
│               ├── AsynchronousMailPersistenceServiceSpec.groovy
│               └── AsynchronousMailServiceSpec.groovy
├── main
│   ├── groovy
│   ├── scripts
│   └── templates
└── test
    └── groovy
        └── grails
            └── plugin
                └── asyncmail
                    ├── AsynchronousMailAttachmentSpec.groovy
                    ├── AsynchronousMailMessageBuilderSpec.groovy
                    ├── AsynchronousMailMessageSpec.groovy
                    ├── AsynchronousMailProcessServiceSpec.groovy
                    ├── CompareMessageBuilderSpec.groovy
                    └── ValidatorSpec.groovy
```

Move Configurations...

Move Configurations

grails-app/conf/BuildConfig.groovy

```
grails.project.work.dir = 'target'
grails.project.source.level = 1.6
grails.project.dependency.resolver = "maven" // or ivy

grails.project.dependency.resolution = {
  inherits 'global'
  log 'warn'
  repositories {
     grailsCentral()
     grailsPlugins()
     grailsHome()

     mavenLocal()
     mavenCentral()
     // uncomment these (or add new ones) to enable remote dependency
resolution from public Maven repositories
     mavenRepo 'http://snapshots.repository.codehaus.org'
     mavenRepo 'http://repository.codehaus.org'
     mavenRepo 'http://download.java.net/maven/2/'
     mavenRepo 'http://repository.jboss.com/maven2/'
 }
  dependencies {
    compile "org.codehaus.gpars:gpars:1.2.1"
  }

  ...
```

build.gradle

```
buildscript {
  ext {
     grailsVersion = project.grailsVersion
  }
  repositories {
     mavenLocal()
     maven { url "https://repo.grails.org/grails/core" }
  }
  dependencies {
     classpath "org.grails:grails-gradle-plugin:$grailsVersion"
  }
}

version "2.0.0.RC2"
group "org.grails.plugins"

apply plugin:"idea"
apply plugin:"org.grails.grails-plugin"
apply plugin:"org.grails.grails-plugin-publish"
apply plugin:"org.grails.grails-gsp"

ext {
  grailsVersion = project.grailsVersion
  gradleWrapperVersion = project.gradleWrapperVersion
}

repositories {
  mavenLocal()
  maven { url "https://repo.grails.org/grails/core" }
}

dependencyManagement {
  imports {
     mavenBom "org.grails:grails-bom:$grailsVersion"
  }
  applyMavenExclusions false
}

...
```

Update plugin version and group

Copy you maven repositories here

Move Configurations…

grails-app/conf/BuildConfig.groovy

```
grails.project.work.dir = 'target'
grails.project.source.level = 1.6
grails.project.dependency.resolver = "maven" // or ivy

grails.project.dependency.resolution = {
  inherits 'global'
  log 'warn'
  repositories {
    grailsCentral()
    grailsPlugins()
    grailsHome()

    mavenLocal()
    mavenCentral()
    // uncomment these (or add new ones) to enable remote dependency resolution from public Maven
repositories
    mavenRepo 'http://snapshots.repository.codehaus.org'
    mavenRepo 'http://repository.codehaus.org'
    mavenRepo 'http://download.java.net/maven/2/'
    mavenRepo 'http://repository.jboss.com/maven2/'
  }

  dependencies {
    compile "org.codehaus.gpars:gpars:1.2.1"
  }

  plugins {
    test(":hibernate:3.6.10.16") {
      export = false
    }
    compile(':mail:1.0.7'){
        excludes 'spring-test'
     }
     compile ':quartz:1.0.2'

    build(':release:3.0.1', ':rest-client-builder:2.0.3') {
      export = false
    }
  }
}
```

Copy plugins under dependencies

build.gradle

```
...
dependencyManagement {
  imports {
    mavenBom "org.grails:grails-bom:$grailsVersion"
  }
  applyMavenExclusions false
}

dependencies {
  compile "org.springframework.boot:spring-boot-starter-logging"
  compile "org.springframework.boot:spring-boot-autoconfigure"
  compile "org.grails:grails-core"
  compile "org.springframework.boot:spring-boot-starter-actuator"
  compile "org.springframework.boot:spring-boot-starter-tomcat"
  compile "org.grails:grails-dependencies"
  compile "org.grails:grails-web-boot"
  compile "org.grails.plugins:cache"
  compile "org.grails.plugins:scaffolding"
  console "org.grails:grails-console"
  profile "org.grails.profiles:web-plugin:$grailsVersion"
  provided "org.grails:grails-plugin-services"
  provided "org.grails:grails-plugin-domain-class"
  testCompile "org.grails:grails-plugin-testing"

  testCompile "org.grails.plugins:hibernate4"

  compile "org.grails.plugins:mail:2.0.0.RC6"
  compile "org.grails.plugins:quartz:2.0.8"
}

task wrapper(type: Wrapper) {
  gradleVersion = gradleWrapperVersion
}

...
```

Move Configurations…

grails-app/conf/BuildConfig.groovy

```
grails.project.work.dir = 'target'
grails.project.source.level = 1.6
grails.project.dependency.resolver = "maven" // or ivy

grails.project.dependency.resolution = {
  inherits 'global'
  log 'warn'
  repositories {
     grailsCentral()
     grailsPlugins()
     grailsHome()

     mavenLocal()
     mavenCentral()
     // uncomment these (or add new ones) to enable remote dependency resolution from public Maven
repositories
     mavenRepo 'http://snapshots.repository.codehaus.org'
     mavenRepo 'http://repository.codehaus.org'
     mavenRepo 'http://download.java.net/maven/2/'
     mavenRepo 'http://repository.jboss.com/maven2/'
  }

  dependencies {
     compile "org.codehaus.gpars:gpars:1.2.1"
  }

  plugins {
     test(":hibernate:3.6.10.16") {
        export = false
     }

     compile(':mail:1.0.7'){
        excludes 'spring-test'
     }
     compile ':quartz:1.0.2'

     build(':release:3.0.1', ':rest-client-builder:2.0.3') {
        export = false
     }
  }
}
```

build.gradle

```
...

task wrapper(type: Wrapper) {
  gradleVersion = gradleWrapperVersion
}
```

Plugin-publish related configurations

```
grailsPublish {
  userOrg = 'kefirsf'
  license {
     name = 'Apache-2.0'
  }
  websiteUrl = 'https://grails.org/plugin/asynchronous-mail'
  issueTrackerUrl = 'https://github.com/kefirfromperm/grails-asynchronous-mail/issues'
  vcsUrl = 'https://github.com/kefirfromperm/grails-asynchronous-mail.git'
  title = "Grails Asynchronous Mail plugin"
  desc = "The plugin realises asynchronous mail sending. It stores messages in a DB and sends
them asynchronously by a quartz job."
  developers = [
        kefirfromperm:"Vitalii Samolovskikh",
        stokito:"Sergey Ponomarev",
        ilopmar:"Iván López",
        burtbeckwith:"Burt Beckwith",
        puneetbehl:"Puneet Behl",
        aberbenni:"Alessandro Berbenni",
        dpcasady:"Danny Casady",
        SAgrawal14:"Shashank Agrawal",
        visheshd:"Vishesh",
        'micke-a':"Mikael Andersson"
  ]
}
```

Move Configurations...

grails-app/conf/DefaultAsyncMailConfig.groovy

```
asynchronous.mail.default.attempt.interval=300000l     // Five minutes
asynchronous.mail.default.max.attempts.count=1
asynchronous.mail.send.repeat.interval=60000l       // One minute
asynchronous.mail.expired.collector.repeat.interval=607000l
asynchronous.mail.messages.at.once=100
asynchronous.mail.send.immediately=true
asynchronous.mail.override=false
asynchronous.mail.clear.after.sent=false
asynchronous.mail.disable=false
asynchronous.mail.useFlushOnSave=true
asynchronous.mail.persistence.provider='hibernate'    // Possible values are 'hibernate', 'hibernate4',
'mongodb'
asynchronous.mail.gparsPoolSize=1
asynchronous.mail.newSessionOnImmediateSend=false
```

application.groovy

```
grails {
  profile = 'web-plugin'
  codegen {
    defaultPackage = 'grails.plugin.asyncmail'
  }
}
info {
  app {
    name = '@info.app.name@'
    version = '@info.app.version@'
    grailsVersion = '@info.app.grailsVersion@'
  }
}
spring {
  groovy {
    template."check-template-location" = false
  }
}
asynchronous.mail.default.attempt.interval=300000l     // Five minutes
asynchronous.mail.default.max.attempts.count=1
asynchronous.mail.send.repeat.interval=60000l       // One minute
asynchronous.mail.expired.collector.repeat.interval=607000l
asynchronous.mail.messages.at.once=100
asynchronous.mail.send.immediately=true
asynchronous.mail.clear.after.sent=false
asynchronous.mail.disable=false
asynchronous.mail.useFlushOnSave=true
asynchronous.mail.persistence.provider='hibernate4'     // Possible values are 'hibernate',
'hibernate4', 'mongodb'
asynchronous.mail.newSessionOnImmediateSend=false

...
```

Move Configurations...

grails-app/conf/DataSource.groovy

```
// environment specific settings
environments {
  test {
    dataSource {
      pooled = true
      driverClassName = "org.h2.Driver"
      username = "sa"
      password = ""
      dbCreate = "create-drop"
      url = "jdbc:h2:mem:testDb;MVCC=TRUE;LOCK_TIMEOUT=10000"
    }
    hibernate {
      cache.use_second_level_cache = false
      cache.use_query_cache = false
    }
  }
}
```

→

application.groovy

```
...
asynchronous.mail.default.attempt.interval=300000l      // Five minutes
asynchronous.mail.default.max.attempts.count=1
asynchronous.mail.send.repeat.interval=60000l       // One minute
asynchronous.mail.expired.collector.repeat.interval=607000l
asynchronous.mail.messages.at.once=100
asynchronous.mail.send.immediately=true
asynchronous.mail.clear.after.sent=false
asynchronous.mail.disable=false
asynchronous.mail.useFlushOnSave=true
asynchronous.mail.persistence.provider='hibernate4'     // Possible values are 'hibernate', 'hibernate4', 'mongodb'
asynchronous.mail.newSessionOnImmediateSend=false
```

```
environments {
  test {
    dataSource {
      pooled = true
      jmxExport = true
      driverClassName = 'org.h2.Driver'
      username = 'sa'
      password = ''
      dbCreate = 'update'
      url =
'jdbc:h2:mem:testDb;MVCC=TRUE;LOCK_TIMEOUT=10000;DB_CLOSE_ON_EXIT=FALSE'
    }
    quartz.jdbcStore = false
  }
}
```

Change Plugin Descriptor file

- Extend the descriptor with **`grails.plugins.Plugin`**
- Refer Plugin descriptor file from new plugin to copy and update all the required methods.
- In addition you should remove the ***“version”*** property from the descriptor as this is now defined in ***“build.gradle”***.

Modify Package Import

- In Grails 3.x all internal APIs can be found in the **org.grails** package.

Modify Package Import

- In Grails 3.x all internal APIs can be found in the **org.grails** package.
- All public facing APIs in the grails package.

Modify Package Import

- In Grails 3.x all internal APIs can be found in the **org.grails** package.
- All public facing APIs in the grails package.
  - The *“`org.codehaus.groovy.grails`”* package no longer exists.

Modify Package Import

- In Grails 3.x all internal APIs can be found in the **org.grails** package.
- All public facing APIs in the grails package.
  - The “`org.codehaus.groovy.grails`” package no longer exists.
  - All package declaration in sources should be modified for new locations of respective classes. For example “`org.codehaus.groovy.grails.commons.GrailsApplication`” is now “`grails.core.GrailsApplication`”

Cleanup

- Delete the files which are no longer used, such as -
  - BuildConfig.groovy
  - Config.groovy
  - DataSource.groovy
  - Old Plugin Descriptor in root etc.

Update Unit Test Cases

```
import grails.test.mixin.TestFor

/**
 * Attachment unit tests
 */
@TestFor(AsynchronousMailAttachment)
class AsynchronousMailAttachmentTests {
    void testDefault(){
        def attachment = new
AsynchronousMailAttachment()
        assertNull attachment.attachmentName
        assertEquals
AsynchronousMailAttachment.DEFAULT_MIME_TYPE,
attachment.mimeType
        assertNull attachment.content
        assertFalse attachment.inline
    }
…
}
```

```
package grails.plugin.asyncmail

import grails.test.mixin.TestFor
import spock.lang.Specification

import static
grails.plugin.asyncmail.AsynchronousMailAttachment.DEFAULT
_MIME_TYPE

@TestFor(AsynchronousMailAttachment)
class AsynchronousMailAttachmentSpec extends Specification
{

    void "testing default constructor"() {
        when:
        AsynchronousMailAttachment attachment = new
AsynchronousMailAttachment()

        then:
        !attachment.attachmentName
        attachment.mimeType == DEFAULT_MIME_TYPE
        !attachment.content
        !attachment.inline
    }
...
}
```

Update Integration Test Cases

```
package grails.plugin.asyncmail

class AsynchronousMailTests extends GroovyTestCase {
    public static final String VALUE_MAIL =
'test@example.com'

    AsynchronousMailService asynchronousMailService;


    void testSendAsynchronousMail(){
        asyncMailService.sendMail {
            to VALUE_MAIL
            subject 'Test'
            text 'Test'

            immediate false
        }

        AsynchronousMailMessage message =
AsynchronousMailMessage.findAll()[0]

        assertEquals(VALUE_MAIL, message.to[0])
        assertEquals(MessageStatus.CREATED, message.status)
    }

}
```

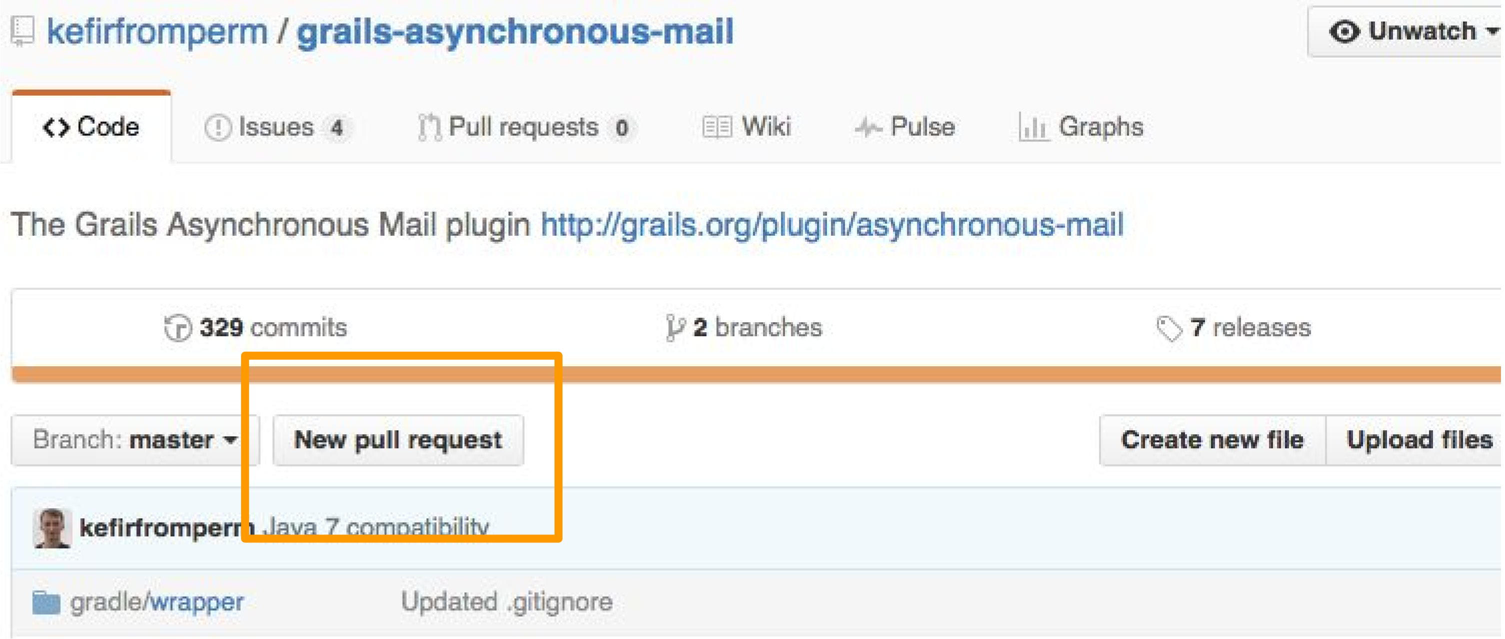
```
package grails.plugin.asyncmail
import grails.test.mixin.integration.Integration
import grails.transaction.Rollback
import org.springframework.beans.factory.annotation.Autowired
import spock.lang.Specification

import static grails.plugin.asyncmail.enums.MessageStatus.CREATED

@Integration
@Rollback
class AsynchronousMailServiceSpec extends Specification {

    public static final String VALUE_MAIL = 'test@example.com'

    @Autowired
    AsynchronousMailService asynchronousMailService

    void testSendAsynchronousMail(){
        when:
        asynchronousMailService.sendMail {
            to VALUE_MAIL
            subject 'Test'
            text 'Test'
            immediate false
        }
        AsynchronousMailMessage message =
AsynchronousMailMessage.findAll()[0]

        then:
        VALUE_MAIL == message.to[0]
        CREATED == message.status
    }
}
```

Checklist Before Sending Pull Request

- ❑ All unused files are removed/deleted.
- ❑ All unused imports are removed.
- ❑ All Unit/Integration test cases are passing.
- ❑ All functional tests are passing.

Now ???

Send Pull Request



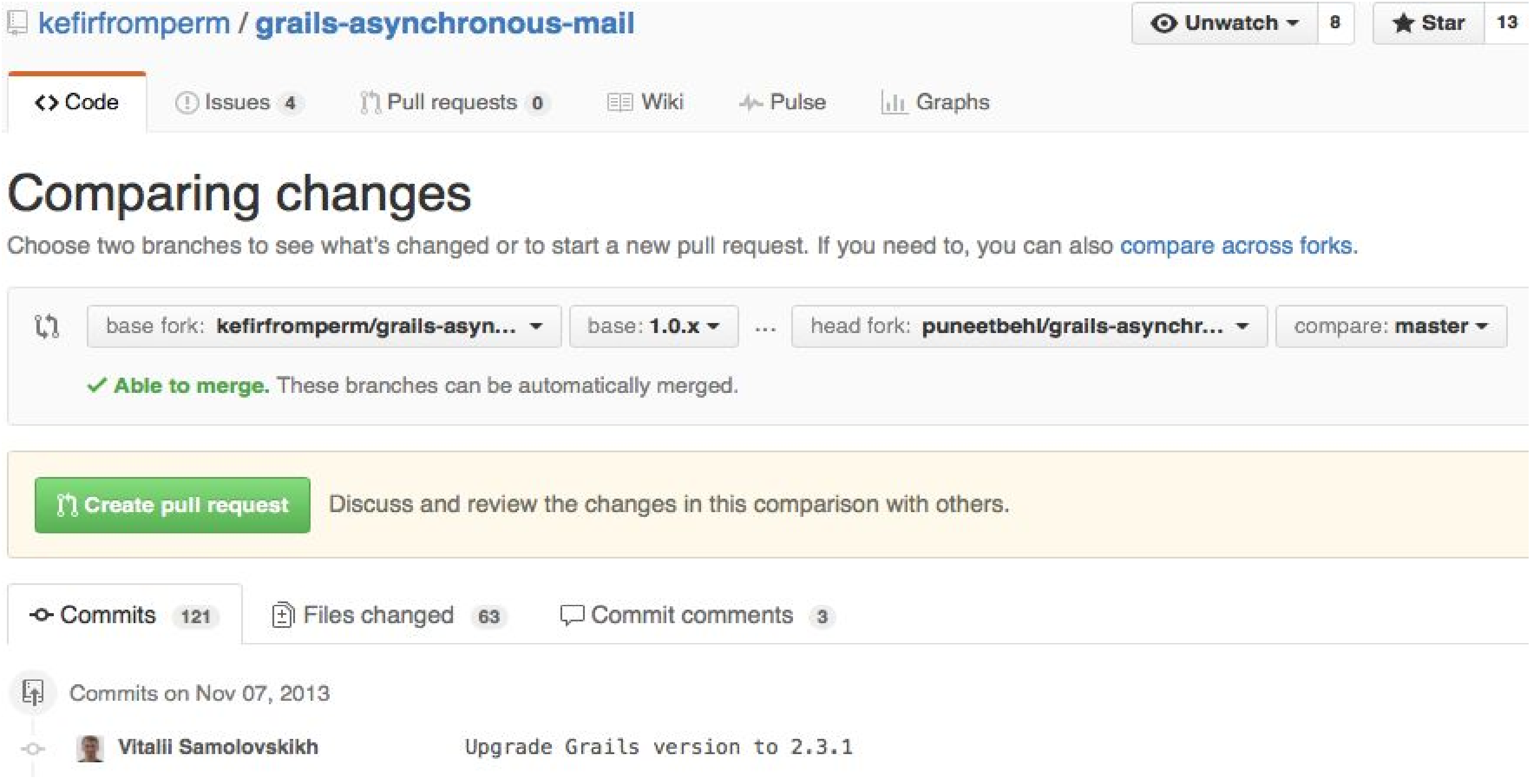
kefirfromperm / grails-asynchronous-mail

The Grails Asynchronous Mail plugin http://grails.org/plugin/asynchronous-mail

kefirfromperm Java 7 compatibility

gradle/wrapper Updated .gitignore

Send Pull Request



kefirfromperm / grails-asynchronous-mail

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also compare across forks.

base fork: kefirfromperm/grails-asyn... base: 1.0.x ... head fork: puneetbehl/grails-asynchr... compare: master

✓ Able to merge. These branches can be automatically merged.

Discuss and review the changes in this comparison with others.

Commits on Nov 07, 2013

Vitalii Samolovskikh Upgrade Grails version to 2.3.1

Next steps???

Next steps...

- Publish plugin
- Update documentation

Publish Plugin

- Signup/Signin to https://bintray.com (one-click signin with GitHub or Twitter)
- Create a new maven repository called plugins on your account
- Edit “`build.gradle`” to customize default configurations
- Now, to public run “`gradle bintrayUpload`”

Publish Plugin

- Signup/Signin to https://bintray.com (one-click signin with GitHub or Twitter)
- Create a new maven repository called plugins on your account
- Edit “`build.gradle`” to customize default configurations
- Now, to public run “`gradle bintrayUpload`”

https://medium.com/@benorama/how-to-publish-your-grails-3-plugin-to-bintray-c341b24f567d#.p8xru1av2

Few useful tips

- **Default plugin configuration** file == `plugin.(yml/groovy)`
- **Travis Integration** - Nicely indicates & reports if new changes and pull request breaks something in test cases
- **Demo Application**
- Goto **Grails slack** #questions & #plugin channel for more help
- Use **`@Commons`** on plugin descriptor class or **`@Slf4j`** `for log support`
- **`grails install`** to install plugin locally and use it direclty in your project

Questions?

Thank you