

ORANGE FT GROUP

# JUnitHistory 1.2

---

## Manuel d'administration

29 mars 2016

---

## Table des matières

Présentation.....	4
Récupération des sources et compilation.....	4
Librairie externe .....	4
Configuration .....	6
Installation et mise à jour .....	6
Base de données .....	6
V1.1 .....	6
V1.2 .....	7
Première installation.....	7
Mise à jour .....	8
Sauvegarde et restauration des fichiers uploades.....	8
Accès à l'application.....	9
Journaux.....	9
Gestion des testeurs .....	11
Edition .....	12
Suppression.....	12
Création.....	13
Gestion des catégories.....	14
Edition .....	15
Suppression.....	15

Création.....	16
Gestion des groupes de STB.....	17
Edition .....	18
Suppression.....	19
Création.....	19

# Présentation

---

Ce document présente la partie installation, paramétrage et administration de l'application JUnitHistory.

Le manuel de l'utilisateur **JUnitHistoryUser.pdf** présente et décrit de manière approfondie les fonctionnalités.

L'application JUnitHistory est une application Web écrite en GWT-RPC.

Elle est installée dans un serveur applicatif java (Tomcat) et utilise une base de données MySQL.

L'interface utilisateur fonctionne sur les navigateurs Firefox et Chrome.

## Récupération des sources et compilation

---

### Librairie externe

La librairie **moxieapps.gwt.uploader** utilisée pour l'upload des rapport doit être récupérée et installée dans le repository maven local.

- ❖ récupérer le jar sur: <http://www.moxiegroup.com/moxieapps/gwt-uploader/>
- ❖ créer l'artifact maven:

```
mvn install:install-file -Dfile=./org.moxieapps.gwt.uploader-1.1.0.jar -DgroupId=org.moxieapps -  
DartifactId=org.moxieapps.gwt.uploader -Dversion=1.1.0 -Dpackaging=jar -DgeneratePom=true
```

**Compilation et packaging**

- ❖ JUnitHistory<profile>.xml (configuration)
- ❖ JUnitHistory<profile>.war (application)

Les sources de l'application sont hébergées dans le Github d'Orange :

```
https://github.com/orange-tv-blagnac/junit-history.git
```

Les récupérer avec `git clone` puis lancer la compilation et la packaging avec maven :

```
mvn clean package -P<profile>
```

Une archive JUnitHistory<profile>.tar.gz est créée.

Elle contient les fichiers:

**Profiles de compilation :**

Pour les équipes externes	Profil par défaut avec tout le paramétrage à réaliser	<code>mvn clean package</code>	junithistory-1.1-SNAPSHOT.tar.gz <ul style="list-style-type: none"> <li>JUnitHistory.war</li> <li>JUnitHistory.xml</li> </ul>
Pour l'équipe NewTV Application de PROD	Profil 'prod' pour déployer l'application de PROD en interne (NewTV). Reste uniquement les mots de passe à paramétrer	<code>mvn clean package -Pprod</code>	junithistory-1.1-SNAPSHOT-prod.tar.gz <ul style="list-style-type: none"> <li>JUnitHistoryProd.war</li> <li>JUnitHistoryProd.xml</li> </ul>
Pour l'équipe NewTV Application de DEV (pour tests ajout fonctionnalites)	Profil 'dev' pour déployer l'application de DEV en interne (NewTV). Reste uniquement les mots de passe à paramétrer	<code>mvn clean package Pdev</code>	junithistory-1.1-SNAPSHOT-dev.tar.gz <ul style="list-style-type: none"> <li>JUnitHistoryDev.war</li> <li>JUnitHistoryDev.xml</li> </ul>

## Configuration

Le fichier JUnitHistory.xml contient les paramétrages de l'application, en ce qui concerne l'accès à la base de données et les credentials des différents du manager et de l'administrateur.

Ces valeurs surchargent les valeurs du fichier /WEB-INF/web.xml inclu dans l'archive.

Les fichiers de configuration sont pré-rempli si on compile avec les profils maven 'prod' ou 'dev' et seuls les mots de passe seront à saisir.

Avec la compilation par défaut, tous les paramètres devront être configurés :

Paramètre	Value
bddUrl	Ip et nom de la base de données
bddUser	Login de la BDD
bddPwd	Password de la BDD
login_admin	Login du profil ' <i>admin</i> ' de l'application
pwd_admin	Pwd du profil ' <i>admin</i> ' de l'application
login_manager	Login du profil ' <i>manager</i> ' de l'application
pwd_manager	Pwd du profil ' <i>manager</i> ' de l'application

## Installation et mise à jour

---

### Base de données

#### V1.1

La base de données **JUnitHistoryTools** doit être créée préalablement à la première installation de l'application.

Le script de création est situé dans les sources du projet :

```
/scripts/JUnitHistoryTools.sql
```

Exécuter ce script dans MySQL / PhpMyAdmin / Nouvelle base de données pour créer une structure vide.

**Important** : pour utiliser une deuxième instance de l'application (DEV), créer une deuxième base de données avec un nom différent (**JUnitHistoryTools\_dev**).

### V1.2

Pour cette nouvelle version une table supplémentaire est nécessaire : tcomment.

Exécuter le patch sql pour mettre à jour la BDD avec cette nouvelle table :

```
/scripts/patch1.1.to1.2
```

## Première installation

Installer si nécessaire un serveur Tomcat

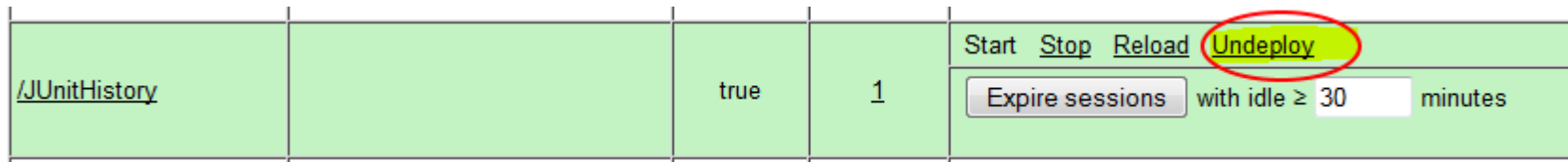
- ❖ Copier le fichier JUnitHistory.xml dans le répertoire /conf/Catalina/localhost de Tomcat
- ❖ Copier le fichier JUnitHistory.war dans le répertoire /webapps/ de Tomcat.

Dans le manager de Tomcat (<http://<IP>:<port>/manager/html>), s'assurer que la ou les instances sont bien installées et démarrées :

<a href="#">/JUnitHistory</a>		true	1	Start Stop Reload Undeploy
				Expire sessions with idle ≥ 30 minutes
<a href="#">/JUnitHistoryDev</a>		true	0	Start Stop Reload Undeploy
				Expire sessions with idle ≥ 30 minutes

## Mise à jour

Désinstaller préalablement la version en cours en utilisant le manager de Tomcat :



Puis réinstaller pour une première installation.

**Remarque** : Il est conseillé d'arrêter Tomcat. Normalement la mise à jour peut se faire à chaud, mais j'ai constaté des anomalies.

## Sauvegarde et restauration des fichiers uploades

Tous les fichiers \*.xml , \*.log, \*.txt uploadés et ajoutés dans l'historique sont stockés dans le répertoire de l'application :

```
/webapps/JUnitHistory/history/xml
```

Une copie de sauvegarde est automatiquement réalisée dans le répertoire :

```
/usr/local/gwtjunithistory/prod
```

(ce path est configuré dans le fichier web.xml – param : historyBackupPath).



Lors d'une mise à jour, le répertoire de l'application est supprimé (undeploy) par Tomcat.

Lors du démarrage de la nouvelle version, l'application constate l'absence du répertoire history et restaure automatiquement les fichiers depuis le répertoire de sauvegarde.

Ces fichiers ne sont pas nécessaires au fonctionnement de l'application puisque toute l'information est stockée en base.

Seuls les fichiers de log sont réellement nécessaires, car ils peuvent être consultés en ligne.

Le maintien des fichiers \*.xml source est néanmoins une sécurité. Ils pourraient permettre de reconstituer intégralement l'historique en cas d'anomalie sur la base de données.

Toutes ces opérations (sauvegarde & restauration) sont automatiques. L'administrateur n'a pas à intervenir.

Il doit cependant s'assurer que le répertoire `/usr/local/gwtjunithistory/` est inclus dans les procédures de sauvegarde du serveur.

## Accès à l'application

L'url dérive directement du nom du fichier war copié dans `/webapps`.

Il est donc conseillé de renommer le war de PROD en JUnitHistory.war pour maintenir les urls connues :

- ❖ url de PROD : `http://<IP> :<port>/JUnitHistory`
- ❖ url de DEV : `http://<IP> :<port>/JUnitHistoryDev`

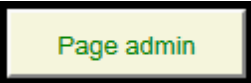
## Journaux

Les journaux sont créés dans le répertoire `/logs` de Tomcat. Un fichier par jour.

Par ex : `junithistory-prod.2016-03-14.log`

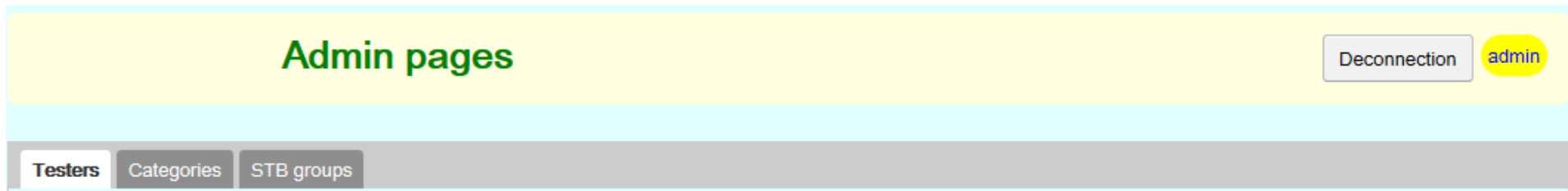
## Pages d'administration

Dans le menu principal de l'application, un utilisateur de profil '*admin*' peut accéder aux pages d'administration en cliquant sur le bouton

Page admin

Trois onglets se présentent

- ❖ Testers
- ❖ Categories
- ❖ STB Groups








## Gestion des testeurs


Ce tableau gère la liste des testeurs.

C'est à partir de cette liste qu'on peut associer un rapport de test et un testeur.

### Testers

id	name	edit	description	admin	del
1	Sylvie F.				
2	Fabrice F.				
3	admin		Pour les taches automatiques		
4	Maxime B.		utilisateur		
5	Anne-Laure E.		utilisateur		

## Edition


En cliquant sur le bouton edit  d'une ligne on ouvre la fenêtre d'édition du testeur:



**Remarque 1** : le testeur 'admin' créé automatiquement n'est pas modifiable

**Remarque2** : le nom du testeur n'est modifiable que s'il n'est lié à aucun rapport.

## Suppression

Le bouton del  d'une ligne permet de supprimer le testeur.

**Remarque 1** : Le testeur 'admin' n'est pas supprimable.

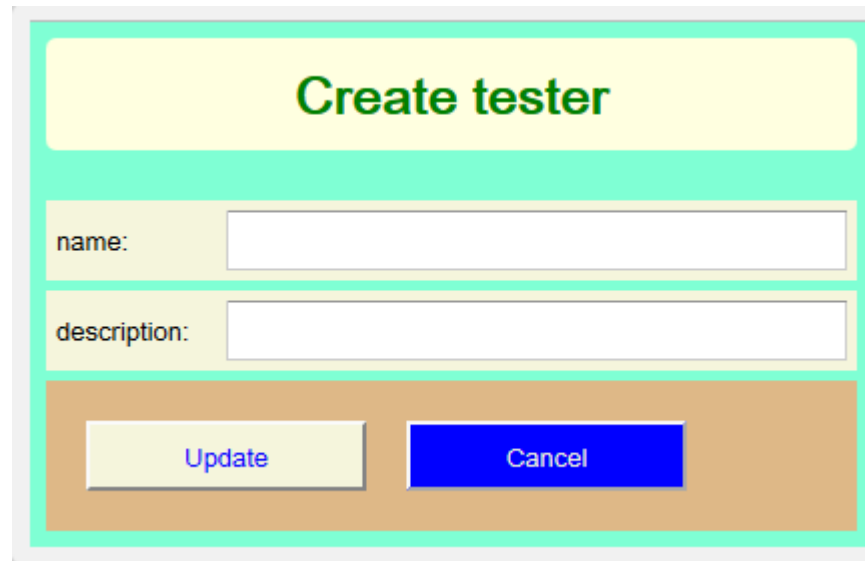
**Remarque 2** : Un testeur ne peut être supprimé que s'il n'est lié à aucune suite de tests.

## Création

Pour créer un nouveau tester, cliquer sur le bouton

Create tester

La fenêtre de création d'un nouveau testeur s'affiche :



Remplir les champs puis cliquer sur

Update









**Remarque** : le nom doit être unique dans la base.

## Gestion des catégories


Les catégories regroupent thématiquement les classes de tests concernant un même domaine fonctionnel.

Ce regroupement est arbitraire et peut être modifié dans cette table.

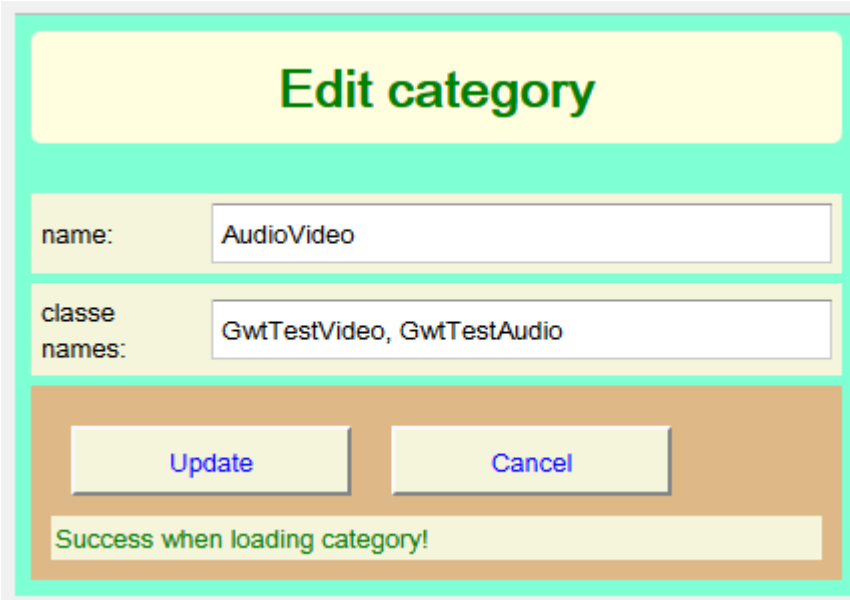
### Categories

id	name	edit	list	default	del
0	Network_wan_lan		GwtTestNetwork, GwtTestLar		
1	Dtt_Dth		GwtTestDtt, GwtTestDth		
2	HDD_Device		GwtTestHDD, GwtTestDevice		
3	AudioVideo		GwtTestVideo, GwtTestAudio		
4	DLNA		GwtTestDmc		
5	TS		GwtTestTimeShift		
6	PVR		GwtTestPVR		
7	other				

## Edition

En cliquant sur le bouton edit  d'une ligne on ouvre la fenêtre d'édition de la catégorie:


(la catégorie 'other' créée automatiquement n'est pas modifiable ni supprimable)



Le nom de la catégorie est unique dans la base.  
Le nom des classes doit être séparé par une virgule.

Cliquer sur le bouton  pour valider.

## Suppression

Le bouton del  d'une ligne permet de supprimer la catégorie.

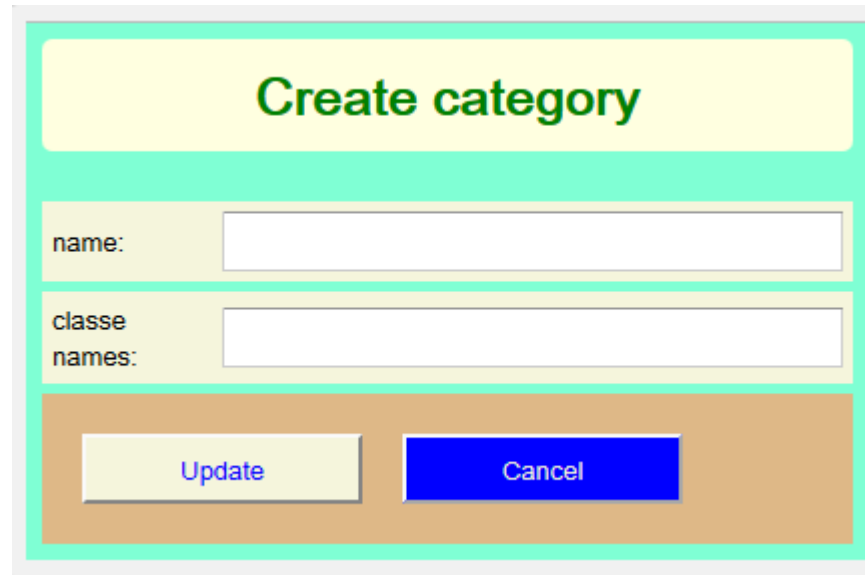
La suppression d'une catégorie n'est possible que si elle n'est référencée par aucune classe de test

## Création

Pour créer une nouvelle catégorie, cliquer sur le bouton

Create category

a fenêtre de création d'une catégorie s'affiche :



Remplir les champs puis cliquer sur

Update

**Remarque** : le nom doit être unique dans la base et le nom des classes doit être séparé par une virgule.










## Gestion des groupes de STB


Ce tableau décrit les STB pour lesquelles on désire maintenir un historique cohérent.

C'est la liste qui apparaît dans la boîte déroulante des pages 'Single report' et 'History reports'.

### STB groups

id	name	edit	stb	prefix	del
2	FrR3-Sagem		Sagem FR	FrR3-sg-	
3	FrR3-Samsung		Samsung FR	FrR3-ss-	
4	FrR4-Sagem		Sagem FR	FrR4-sg-	
5	FrR4-Samsung		Samsung FR	FrR4-ss-	
8	Newbox-RQS5		Newbox	NBXR2-	
10	Newbox-R4		Newbox	NBXR4-	
11	MIB4		MIB4	MIB4-	

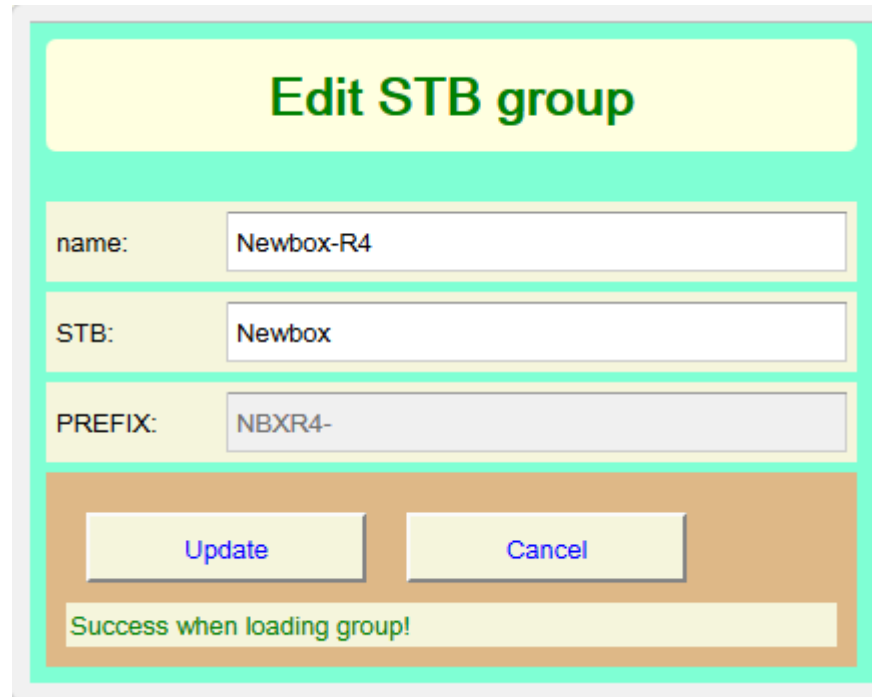
## Edition

En cliquant sur le bouton edit  d'une ligne on ouvre la fenêtre d'édition de STB:

**Name** : nom du groupe (STB + projet)

**STB** : boitier physique

**Prefix** : utilisé pour préfixer les fichiers uploadés.




Le nom du groupe est unique dans la base.

Le préfixe doit se terminer par un tiret.

Cliquer sur le bouton pour valider.

**Remarque** : le préfixe n'est modifiable que dans un groupe non référencé par un test existant.

### Suppression

Le bouton del  d'une ligne permet de supprimer le groupe de STB.

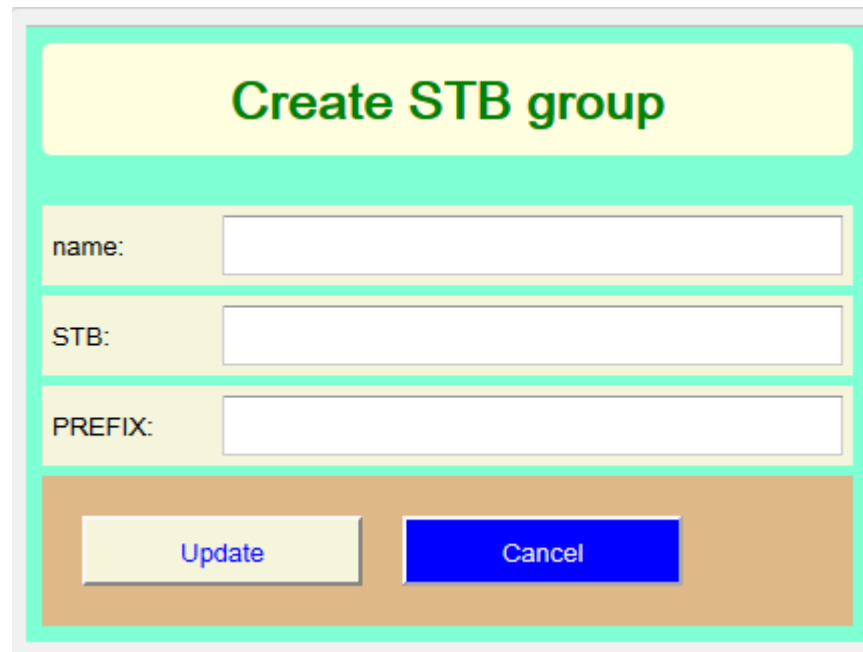
La suppression du groupe n'est possible que s'il n'est référencé par aucun rapport.

### Création

Pour créer un nouveau groupe de STB, cliquer sur le bouton

Create STB group

La fenêtre de création d'un groupe s'affiche :



Remplir les champs puis cliquer sur

Update

**Remarque** : le nom du groupe et le préfixe doivent être uniques dans la base.