

июн. 16, 2024

Содержание

1	Обзор	2
1.1	Назначение	2
1.2	Установка	2
1.3	Добавление проекта в рабочее пространство	3
1.4	Работа с активным проектом	4
1.5	Обновление активного проекта	4
1.6	Конфигурирование сервера приложения	4
1.7	Изменения настроек проекта	4
1.8	Изменения глобальных настроек	4
1.9	Хранения паролей	4
1.10	Горячие клавиши	4
2	Настройки	5
2.1	Настройки окружения	5
2.2	Настройки проекта	5
3	Ярлыки	6
3.1	activate_project.cmd	6
3.2	active_project_configure_idea.cmd	6
3.3	active_project_refresh.cmd	7
3.4	active_project_sbt.cmd	7
3.5	active_project_start_idea.cmd	7
3.6	start_sep_idea.cmd	7
3.7	add_project.cmd	7
3.8	delete_project.cmd	7
4	Непрерывная интеграция	7
4.1	Настройка в linux	7
4.2	Сборка проекта	8
4.3	Сборка релиза	8
4.4	Сборка snapshot	8
5	Конфигуратор проектов	9
5.1	Commands:	9
6	Менеджер проектов	11
6.1	Commands:	11

7	Менеджер учетных данных	15
7.1	Commands:	15
8	Утилита для git	16
8.1	Commands:	16

1 Обзор

1.1 Назначение

Командная утилита предназначена для автоматизации работы разработчика.

Gsf-cli позволяет:

- Подготовить прикладной проект к работе
- Обновить зависимости необходимые для работы проекта

1.2 Установка

1. Скачайте дистрибутив `gsf-cli`

Внимание: `curl https://repo.global-system.ru/artifactory/common/ru/bitec/gsf-cli/SNAPSHOT/gsf-cli-SNAPSHOT.zip -output gsf-cli.zip`

Для ручного обновления утилиты можно в каталоге `c:\programs\` сделать cmd файл следующего содержания (пути подправить по необходимости) `curl https://repo.global-system.ru/artifactory/common/ru/bitec/gsf-cli/SNAPSHOT/gsf-cli-SNAPSHOT.zip -output gsf-cli.zip «C:\Program Files\7-Zip\7z.exe» x gsf-cli.zip -aoa -ogsf-cli pause`

2. Распакуйте архив
Рекомендуемый путь для установки: `c:\programs\gsf-cli`
3. Установите jdk
Установленные jdk будут искать по адресу `c:\Program Files\Java`
4. Установите IntelliJ Idea
Установленные среды будут искать по адресу `C:\Program Files\JetBrains`

Внимание: В Idea должен быть установлен плагин scala. Подробности корректной установки смотрите в руководстве прикладного разработчика Global3 Framework.

5. Установите `sbt` версии 1.8.2 или выше

Внимание: Sbt должен быть установлен по адресу `c:\programs\sbt\`.

6. При необходимости установите svn клиент

Для авто поиска путь доступа к `svn.exe` должен быть добавлен с системную переменную `path`.
Установщик TortoiseSVN может делать это автоматически

7. При необходимости установите git клиент

1.3 Добавление проекта в рабочее пространство

Внимание: Если перед началом работы открыта среда разработки в общем окружении ее необходимо закрыть.

Для добавление проекта запустите скрипт `gsf-cli\links\add_project.cmd` и следуйте инструкциям мастера. Мастер запросит необходимые параметры, и проведет подготовку проекта к работе.

Внимание: При возникновении ошибки загрузки модулей из GitLab fatal: Unencrypted HTTP is not supported for GitHub. Ensure the repository remote URL is using HTTPS. следует выполнить команду `git config --global credential.extgit.global-system.ru.provider generic` и повторить добавление проекта.

Результат выполнения всех шагов мастера:

- `gsf-cli\workspace\dists\{project_name}\Global3se\`
Актуальный дистрибутив сервера приложения
- `gsf-cli\workspace\sources\{project_name}\application\`
Полностью готовый к работе проект с исходным кодом
- `gsf-cli\workspace\links\{project_name}\`
Ярлыки быстрого запуска
- добавленный проект становится активным

Источник проекта

Мастер конфигурации запрашивает источник определяющий откуда будет получен исходный код проекта. Формат источников:

- svn

`https://{path}/application`

- git

`https://{path}.git`

- lxc

`lxc://{host}`

Lxc является контейнером в котором собирается проект в системе CI

1.4 Работа с активным проектом

Активный проект это проект который будет использоваться по умолчанию в случаи если он не указан явно. Часты команды по работе с активным проектом смотрите в `gsf-cli\links\`

1.5 Обновление активного проекта

Для обновления зависимостей активного проекта запустите `gsf-cli\links\active_project_refresh.cmd`

1.6 Конфигурирование сервера приложения

Сервер приложения конфигурируется автоматически, для этого используется профиль конфигурации. Пример: `http://svn.bitec.ru/svn/depot/ASSource/database/pgtest/application/project/deploy/dev-win`

1.7 Изменения настроек проекта

- Удалите проект командой `gsf-cli\links\delete_project.cmd`
При вопросе об удалении файлов ответьте `нет`
- Добавьте проект с тем же именем и новыми параметрами

Примечание: Данный подход имеет смысл только в случаи если не меняется источник проекта. Это позволяет избежать повторной выгрузки и компиляции проекта

1.8 Изменения глобальных настроек

Для изменения глобальных `cli` запустите в консоли команду `gsf-cli\config.cmd configure`

1.9 Хранения паролей

Пароли сохраняются в зашифрованном виде по мастер ключу. Мастер ключ создается автоматически при первом добавлении проекта. Мастер добавления проекта запрашивает необходимые для дальнейшей работы пароли. Для изменения паролей смотри раздел `credential_manager`

1.10 Горячие клавиши

- **вверх, вниз**
Используется для выбора разных вариантах
- **вправо**
Используется для авто завершения команды

2 Настройки

2.1 Настройки окружения

Путь к мастер ключу

Мастер ключ генерируется при первом запуске проекта, и используется для шифрование настроек проекта. Мастер ключ должен находится в безопасном месте и быть не доступным для других пользователей. По умолчанию мастер ключ сохраняется в рабочем каталоге пользователя.

Путь к IntelliJ Idea

Используется для запуска среды

Путь к svn

Используется для работы с `svn`

Путь к sbt

Используется для работы с `sbt`.

Примечание: Sbt версии 1.8 для работы в режиме bsp требует отсутствие пробелов в пути. В связи с этим на данный момент требуется устанавливать sbt по адресу: `c:\programs\sbt`

Начало диапазона динамических портов

Используется для динамического выделения портов, при добавлении проекта с нестандартными портами. Позволяет одновременно запускать несколько серверов приложений.

2.2 Настройки проекта

В данной главе описываются параметры которые может спрашивать мастер, для корректного конфигурирования проектов.

Jdk

Jdk с которым будет работать проект

Url к проекту

Исходный код проекта в svn. Пример: `http://svn.bitec.ru/svn/depot/ASSource/database/pgtest/application`

Url к серверу приложения

Место откуда брать обновления для сервера приложения. Пример: `ftp://ftp.bitec.ru/pub/#Global/Global3/release/Postgres/artifacts/globalserver.zip`

Использовать стандартные порты

Если флаг сброшен, то при конфигурировании правила запуска сервера приложения порты будут динамически выделены из диапазона.

Примечание: Номера портов распечатываются при добавлении проекта. Так же их можно посмотреть в `workspace\sources\{project_name}\application\.idea\runConfigurations\Global3se.xml`

Флаг сборки релиза

По умолчанию сброшен. Если флаг установлен сборка проекта идет в режиме релиза. Что означает что сборка запустится один раз на версию. Повторные запуски будут игнорироваться. Повторная публикация артефактов релиза запрещена. Для смены параметра смотри: `manage.py set_is_publish_release [-h]`

3 Ярлыки

Ярлыки используются для быстрого запуска часто используемых команд. Скрипты для ярлыков находятся по адресу: `gsf-cli\links`

3.1 activate_project.cmd

Активировать проект. При запуске скрипта откроется мастер позволяющий выбрать проект для активации.

3.2 active_project_configure_idea.cmd

Сконфигурировать idea для активного проекта.

3.3 active_project_refresh.cmd

Обновить зависимости для активного проекта.

3.4 active_project_sbt.cmd

Запустить консоль sbt для активного проекта.

3.5 active_project_start_idea.cmd

Запустить среду разработки в общем окружении для активного проекта.

Внимание: Внимание в один момент времени может быть запущена только одна среда разработки в общем окружении.

3.6 start_sep_idea.cmd

Запустить среду разработки в отдельном окружении. Это позволяет запускать несколько сред разработки одновременно. При запуске скрипта мастер запросит проект для запуска. Рабочее окружение для работы idea сохраняется по адресу: `gsf-cli\workspace\idea\{project_name}`

3.7 add_project.cmd

Добавить проект. При этом откроется консоль с мастером для добавления проекта.

3.8 delete_project.cmd

Удалить проект. При этом откроется консоль с мастером для выбора и удаления проекта.

4 Непрерывная интеграция

Gsf-cli может быть встроена в конвейер непрерывной интеграции для публикации решений и комплектов сборки.

4.1 Настройка в linux

1. Возьмите исходники из git
2. Установите необходимые пакеты

```
sudo bin/installpkg.sh
```

3. Установите Gsf-cli

```
bin/install.sh
```

4. Переведите утилиту в автономный режим

```
./config.sh enable_headless
```

5. Загрузите конфигурацию

```
./config.sh load -f config.json
```

6. Запустите сборку

```
./manage.sh --all build
```

4.2 Сборка проекта

Алгоритм сборки:

1. Если необходимо, загрузить исходный код проекта.
2. Если необходимо, загрузить сервер приложения
Сервер приложения будет загружен либо из конфигурации проекта либо из настройки проекта. Настройка проекта может перекрывать конфигурацию в случае если сервер берется не из комплекта сборки. Если решение собирается из комплекта сборки перекрытие запрещено так как это может привести к конфликтам.
3. Если необходимо, загрузить плагин
4. Скомпилировать проект
5. Если необходимо опубликовать артефакты
Необходимость публикации определяется настройками в `project.yaml` и конфигурацией `gsf-cli`

4.3 Сборка релиза

Релиз собирается только 1 раз, после успешной сборки повторный запуск игнорирует изменения до тех пор пока не изменится версия релиза.

Это гарантирует неизменность артефактов в репозитории. В случае если допустить изменяемость релизов в репозитории невозможно гарантировать корректную доставку артефактов так как maven загружает релиз только 1 раз и далее не проверяет его на изменения.

4.4 Сборка snapshot

Snapshot версия собирается и публикуется на каждый запуск.

Совет: Для обновления перевыпущенных артефактов воспользуйтесь командой `sbt update; updateClassifiers` Подробнее смотрите [управления зависимостями в sbt](#)

5 Конфигуратор проектов

Для запуска используйте `gsf-cli\config.cmd`. Используется для расширенного конфигурирования утилиты в случае если не хватает ярлыков.

5.1 Commands:

```
usage: config.py [-h] cmd ...

positional arguments:
  cmd                  Команды
  full_help            Распечатать справку
  configure            Обновить конфигурацию
  load_config          Загрузить конфигурацию
  add_project          добавить проект
  delete_project       Удалить проект
  activate_project     Активировать проект
  enable_headless      Включить автономный режим
  disable_headless     Выключить автономный режим

options:
  -h, --help          show this help message and exit
```

Full_help

```
usage: config.py full_help [-h]

options:
  -h, --help  show this help message and exit
```

Configure

```
usage: config.py configure [-h]
```

options:

-h, --help show this help message and exit

Load_config

```
usage: config.py load_config [-h] [-f F]
```

Загружает конфигурацию из файла.

Конфигурация проекта содержит json файл с атрибутами:

sbt_home - местоположение sbt, если не задан sbt ищется из переменной окружения path

svn_path - местоположение svn, если не задано svn ищется из переменной окружения path

projects - массив проектов.

Атрибуты проекта:

name - имя проекта

project_source - источник проекта

jdk_home - адрес локации jdk

server_source - источник сервера приложения, игнорируется если сборка проекта идет от  комплекта сборки

options:

-h, --help show this help message and exit

-f F файл конфигурации

Add_project

```
usage: config.py add_project [-h]
```

Добавляет проект, конфигурация задается мастером создания проекта

options:

-h, --help show this help message and exit

Delete_project

```
usage: config.py delete_project [-h]
```

Мастер удаления проекта из конфигурации

options:

-h, --help show this help message and exit

Activate_project

```
usage: config.py activate_project [-h]

options:
  -h, --help  show this help message and exit
```

Enable_headless

```
usage: config.py enable_headless [-h]

А автономном режиме запрещено взаимодействие с пользователем.
В случаи необходимости запроса пользователя будет выброшено исключение

options:
  -h, --help  show this help message and exit
```

Disable_headless

```
usage: config.py disable_headless [-h]

А интерактивном режиме возможно взаимодействие с пользователем

options:
  -h, --help  show this help message and exit
```

6 Менеджер проектов

Для запуска используйте `gsf-cli\manage.cmd`. Используется для расширенного управления проектами в случаи если не хватает ярлыков.

6.1 Commands:

```
usage: manage.py [-h] [-p P] [--all] cmd ...

positional arguments:
  cmd                  Команды
  full_help            Распечатать справку
  prepare_project      Подготовить проект к работе
  refresh_server       Обновить сервер приложения
  refresh_source       Обновить исходный код
  refresh              Обновить зависимости
  init_project         Инициализировать проект проекта
  configure_idea       Настроить idea
  set_is_publish_release Установить признак публикации релиза
```

(continues on next page)

(продолжение с предыдущей страницы)

refresh_links	Обновить ярлыки
publish	Опубликовать
publish_sbt_plugin	Опубликовать sbt plugin
build	Собрать проект
clean	Очистить
options:	
-h, --help	show this help message and exit
-p P	Имя проекта
--all	Выполнить действие для всех проектов

Full_help

```
usage: manage.py full_help [-h]
```

options:

- h, --help show this help message and exit

Prepare_project

usage: manage.py prepare_project [-h]
Подготавливает проект к работе, загружает сервер приложения, исходный кода, а так же  конфигурирует idea
options:
-h, --help show this help message and exit

Refresh_server

usage: manage.py refresh_server [-h]	
Обновляет сервер приложение	
options:	
-h, --help	show this help message and exit

Refresh_source

usage: manage.py refresh_source [-h]	
Обновляет исходный код проекта, при необходимости делает checkout проекта	
options:	
-h, --help	show this help message and exit

Refresh

```
usage: manage.py refresh [-h]
```

Обновляет зависимости

options:

-h, --help show this help message and exit

Init_project

```
usage: manage.py init_project [-h]
```

Инициализация проекта, создание необходимых файлов перед запуском idea

options:

-h, --help show this help message and exit

Configure_idea

```
usage: manage.py configure_idea [-h]
```

Конфигурация idea.

При этом происходит:

Создание конфигурации для запуска сервера приложения;

Настройка для проектов системы контроля версий.

Смотри IntelliJ Idea: Settings > Version Control > Directory mappings

options:

-h, --help show this help message and exit

Set_is_publish_release

```
usage: manage.py set_is_publish_release [-h]
```

Вызывает мастера установки признака публикации релиза.

В случае если признак установлен публикация происходит по версии заданной в конфигурации проекта.

options:

-h, --help show this help message and exit

Refresh_links

```
usage: manage.py refresh_links [-h]
```

Обновляет ярлыки

options:

-h, --help show this help message and exit

Publish

```
usage: manage.py publish [-h]
```

Опубликовать комплект сборки

options:

-h, --help show this help message and exit

Publish_sbt_plugin

```
usage: manage.py publish_sbt_plugin [-h]
```

Опубликовать sbt plugin из комплекта сборки

options:

-h, --help show this help message and exit

Build

```
usage: manage.py build [-h]
```

Выполняет обновление сервера, плагина, компиляцию и публикацию

options:

-h, --help show this help message and exit

Clean

```
usage: manage.py clean [-h]
```

Очистить

options:

-h, --help show this help message and exit

7 Менеджер учетных данных

Для запуска используйте `gsf-cli\credential_manager.cmd`. Используется для управление паролями для доступа к репозиторию

7.1 Commands:

```
usage: credential_manager.py [-h] cmd ...

positional arguments:
  cmd          Команды
  full_help    Распечатать справку
  get          Загрузить конфигурацию
  show         Отобразить учетные данные
  set          Задать учетные данные по протоколу

options:
  -h, --help  show this help message and exit
```

Full_help

```
usage: credential_manager.py full_help [-h]

options:
  -h, --help  show this help message and exit
```

Get

```
usage: credential_manager.py get [-h] [-u U]

Получает учетные данные для заданного url.
Учетные данные предоставляются в поток вывода в формате json

options:
  -h, --help  show this help message and exit
  -u U        url для которого надо получить учетные данные
```

Show

```
usage: credential_manager.py show [-h]

Отображает сохраненные учетные данные

options:
  -h, --help  show this help message and exit
```

Set

```
usage: credential_manager.py set [-h] [-u U] [-l L] [-p P]
```

Задаёт учетные данные для заданного url, поиск будет идти по началу строки.

options:

-h, --help	show this help message and exit
-u U	Url для которого надо задать учетные данные
-l L	Имя пользователя
-p P	Пароль

8 Утилита для git

Используется для автоматизации выпуска релизов и переключения между ветками в проекте решения.

Для запуска используйте `gsf-cli\gsf_git.cmd`.

Совет: Для удобной работы из консоли решения используйте алиас `gsfp_git.cmd`.

Консоль решения можно открыть ярлыком `workspace/links/[project_name]/start_shell.cmd`

Алгоритм выпуска релизов:

1. Откройте консоль решения для выпуска релиза
2. Выпустите релиз модуля
В каталоге модуля выполните `gsfp_git create_release`
3. Откройте консоль решения для релиза

Совет: Для переключения решения в релизную ветку в каталоге решения выполните `gsfp_git switch release`

4. Переключите релиз
Для этого в каталоге модуля выполните `gsfp_git switch [release_name]`
5. Сделайте `commit` и `push` для проекта решения

8.1 Commands:

```
usage: gsf_git.py [-h] [-p P] [-w W] [-m M] [-s] [--all_modules] cmd ...
```

positional arguments:

cmd	Команды
full_help	Распечатать справку
refresh	Обновить исходный код решения из git репозитория
create_release	Выпустить релиз
switch	Переключится на другую ветку

(continues on next page)

```
status      Отобразить статус
version_info Отобразить информацию по версиям
```

options:

```
-h, --help      show this help message and exit
-p P            Имя проекта
-w W            Рабочий каталог
-m M            Имя модуля
-s              Решение
--all_modules   Все модули
```

Full_help

```
usage: gsf_git.py full_help [-h]
```

options:

```
-h, --help  show this help message and exit
```

Refresh

```
usage: gsf_git.py refresh [-h]
```

Обновляет исходный код решения и модулей git

options:

```
-h, --help  show this help message and exit
```

Create_release

```
usage: gsf_git.py create_release [-h]
```

Выпускает релиз.

При этом:

- Происходит создание тегов по текущим версиям модулей
- Увеличивается текущая версия модуля
- Происходить commit и push изменений и тега

options:

```
-h, --help  show this help message and exit
```

Switch

```
usage: gsf_git.py switch [-h] branch
```

Переключает репозиторий на другую ветку.

При запуске от решения:

- Изменяется ветка в настройках проекта
- Происходит обновление репозитория

При запуске от модуля:

- Изменяется ветка в файле `project.yaml`
Внимание: Данные изменения не попадают в commit
- Происходит обновление исходного кода по модулю

positional arguments:

branch

options:

-h, --help show this help message and exit

Status

```
usage: gsf_git.py status [-h]
```

Отображает информацию о состоянии решения и модулей.

Позволяет увидеть список модулей по которым необходимо сделать commit или push.

Решение в списке обозначено символом `.`

options:

-h, --help show this help message and exit

Version_info

```
usage: gsf_git.py version_info [-h]
```

Отображает информацию по версиям решения и модулей.

Решение обозначается символом `.`

options:

-h, --help show this help message and exit