

Сервисы

- [Система взаимодействия с СУБД](#)
- [Супервизор и система пакетов](#)
- [Система развертывания СУБД](#)
- [Создание нового хоста \(пакета\)](#)

Система взаимодействия с СУБД

DbQueryPool

DB Query Pool формирует специальный пул для быстрого взаимодействия с множеством клиентов.

Каждый клиент резервирует готовое соединение с базой на момент выполнения запроса. При этом в пуле существует ограничение количества возможных занимаемых сессий на пользователя, формируется очередь выполнения запросов, если пул переполнен.

DbTransactions

DB Transactions организывает способ работы с транзакцией в отдельном соединении, с возможностью ее передачи между любыми сервисами.

Соответственно сам DB Transaction организывает соединение на каждую транзакцию и завершает это соединение по команде сервиса (или таймаутом).

DbMigrator

DBMigrator позволяет следить и модернизировать версии базы. Это необходимо для поддержания актуальной структуры базы в случае обновлений.

DbLoader

DBLoader позволяет загружать сущности, такие как функции или типы (зависит от базы данных) и поддержания их в актуальной версии, через отслеживание их хеша.

DbServices

DbServices позволяет описывать сервисы, расположенные в базе данных, которые реализуют вызовы с определенным, продуманным интерфейсом, и не зависящим от типа базы.

Также это помогает определить, какие сервисы доступны в текущей системе и тестировать их функционал.

Супервизор и система пакетов

Это специальный хост, который запускает другие хосты для организации автономной системы, которая может работать без использования внешних систем управления процессами.

- Запуск и отслеживание состояния, пакетов, перехват вывода управляемых процессов
- Указание общего конфигурационного файла
- Установка и удаление пакетов вручную
- Проверка обновлений пакетов через сервер обновления CarabiSolutions, получение новых пакетов

Пакеты

Внутри супервизор оперирует Пакетами Супервизора - это Node приложение, особой структуры, распространяемое в виде архива.

Сервисы супервизора

Microservice

Microservice это специальная библиотека позволяющая быстро создать первоначальную инфраструктуру хоста для связи с основными системами основными компонентами системы и упрощающую разработки новых хостов.

Система развертывания СУБД

Система установки базы состоит из четырех частей, которые запускаются последовательно, до запуска основных сервисов. Их можно использовать для установки различных компонентов в базу. Чтобы использовать установщики, надо добавить в хост определенный набор сервисов. При запуске хоста эти сервисы будут сканировать директории, которые располагаются внутри пакета (хоста), выполнят подключение к базе через DbTransactions и совершать установку, а также зафиксируют текущее состояние установки.

OraMigrator

OraMigrator работает по принципу последовательной установки изменений (миграций, патчей) базы. Каждое изменение должно иметь определенный номер (версию), определяющую порядок их установки. Каждое изменение выполняется однократно за всю историю базы. OraMigrator выполняет любые SQL запросы без изменений.

Сохранять изменения необходимо в директории `./ora-migrations`. Внутри этой директории можно расположить файлы двумя способами:

- Название файла должна соответствовать версии миграции `{№миграции}.sql`.
- Как альтернатива, можно создать директорию `{№миграции}`, внутри которой разместить файл `index.sql`.

Внутри `index.sql` можно перечислить файлы в этой директории, которые будут выполняться последовательно. Формат `index.sql` принимает только имена файлов, либо комментарии:

```
@MY_FILE_1.sql;  
-- Это комментарий  
@MY_FILE_2.typ;
```

А формат дополнительных файлов соответствует одиночному SQL файлу. Символ `\` **на отдельной строке** разделяет запросы:

```
SELECT ORA_DATABASE_NAME FROM DUAL  
  
\  
SELECT ORA_DATABASE_NAME FROM DUAL
```

В конце оставлять символ `\` в конце файла SQL **запрещено**

Текущее состояние базы OraMigrator хранит внутри Oracle, в таблице `MIGRATIONS`

В случае успешного выполнения миграции он заполнит новую запись об успехе. Она предотвращает повторное выполнение этого изменения.

В случае ошибки выполнения SQL, будет сделана запись с ошибкой, с указанием текста самой ошибки (колонок `ERROR`) и работа OraMigrator прервется. Повторный запуск OraMigrator найдет в таблице ошибку мигратор и сразу прервется, дабы не нарушить дальнейшую работу мигратора. После устранения ошибки надо изменить эту запись вручную и перезапустить хост.

Чтобы не подключаться к Oracle напрямую, можно, через Админ панель, использовать сервис `OraMigratorAdmin`:

- `fixError`, если **не требуется** повторное выполнение миграцией с ошибкой
- `delError`, если **требуется** ее повторное выполнение

PkgInstaller

Если OraMigrator предназначен для выполнения любых SQL-запросов, PkgInstaller устанавливает в Oracle такие сущности, как типы, пакеты, структуры, функции и процедуры, сверяя актуальность их версий, и не выполняя повторную установку, если версия актуальна. PkgInstaller сравнивает версии с помощью вычисления хэш-значения содержимого файла по установке сущности с записью в таблице `PACKAGE_VERSIONS`, которая осталась после предыдущей установки.

В случае необходимости принудительной переустановки сущности, необходимо удалить предыдущую запись или изменить ее хэш-значение в колонке `package_hash`.

Установочные SQL файлы необходимо размещать в директориях с определенным расширением в зависимости от типа сущности:

- Пакеты: `./ora-packages/*.pck`
- Типы: `./ora-types/*.typ`

После установки всех новых пакетов PkgInstaller запустит компиляцию новых пакетов автоматически.

DBServiceLoader

DB-сервисы это интерфейс, схожий с сервисами, позволяющий выполнять функции внутри баз данных, абстрагируясь от вида базы данных. Каждый DB-сервис содержит методы, которые будут выполнить напрямую внутри базы данных.

Создать, ознакомиться с существующими DB-сервисами, и протестировать их, можно через админ панель.

DBServiceLoader позволяет автоматически загружать DB-сервисы из файлов которые размещены в директории `./db-services/*.json`. Формат файла - JSON, который можно скачать в админ панели (предварительно создав в ней DbService вручную).

Общий список DB-сервисов формируется каждый раз при запуске, в памяти сервиса `DbServices`, и нигде не хранится.

DockindInstaller

Его задача загрузить через ядро информационных объектов ее сущности.

DockindInstaller использует DB-сервис `PKG_XML_REPL_CS` без привязки к базе для своей работы.

Контроль версии происходит по принципу PkgInstaller. Вычисляется хэш содержимого файла и сравнивается с хешом внутри ядра (с таблицей внутри базы данных Oracle `DOCKIND_VERSIONS`).

Внутри директории нужно расположить поддиректории, совпадающей с названием категории, в которой они будут размещены в браузере ИО. В каждой поддиректории-категории нужно расположить файл `category.json`, описывающий эту категорию:

```
{
  "name": "Документация по модулям",
  "sysname": "HELP",
  "roles": ["Administrator", "Developer"]
}
```

Там же, необходимо разместить файлы с определенным названием, которые совпадают с названием типа информационного объекта:

- `{dockind}_vocabs.xml` - Словари для типа ИО dockind
- `{dockind}_types.xml` - Определение типа ИО dockind

Например, для категории `HELP` с объектами `HELP_DOC`, `SECOND`

- `./dockinds/HELP/category.json`
- `./dockinds/HELP/HELP_DOC_vocabs.xml`
- `./dockinds/HELP/HELP_DOC_types.xml`
- `./dockinds/HELP/SECOND_vocabs.xml`
- `./dockinds/HELP/SECOND_types.xml`

Создание нового хоста (пакета)

Копирование шаблона

Настройка новый хост

Создание схемы сервиса

Создание сервиса

Подключение сторонних сервисов

Размещение в репозитории

Настройка системы обновления

1. Зайти в <https://carabi.csp.carabisol.ru/>
2. Выбрать на рабочем столе "Пакеты"
3. Создать новый пакет, заполнить "Наименование" (PKG_NAME в build.sh), "Версия" (совпадает с веткой в GIT)
4. Сохранить

Подключение автосборщика

1. Зайти в автосборщик
2. "Добавить" слева снизу
3. Id - (PKG_NAME в build.sh)
4. Url - Url из GitTea
5. Branch совпадает с веткой в GIT
6. Execs - **/bin/bash ./makepkg.sh <token системы обновления>**
7. Сохраняйте и нажмите "Проверка репозитория"
8. Если все успешно - в системе обновления, в пакете, во вкладке "Релизы", появится первый релиз

Тестирование

1. в системе обновления на рабочем столе "Супервизоры"
2. Выбрать супервизор для теста
3. В "**Пакеты**" добавить новый пакет (не в релизах, а именно "Пакеты")
4. Сохранить супервизор
5. Зайти в админку этого супервизора, либо дождаться автообновления
6. Там в разделе "Пакеты" нажать "Получить обновления"