

RoadMap

Основные ссылки

[установка и настройка symfony](#)

[установить скуп](#)

[установить симфони](#)

[контроллеры и маршруты 1](#)

[контроллеры и маршруты 2](#)

[настройка бд через докер](#)

при создании миграции, если есть ошибка, что не может найти драйвер pgsql, нужно зайти в php.ini нужно раскомментировать (убрать ;) ;extension=pgsql и ;extension=pdo_pgsql

[создание, запуск, откат миграций, изменение таблиц query Builder](#)

[отличие дат](#)

[сериалайзер, чтобы объекты отдавать пользователю](#)

[отладка приложения](#)

[мастер установки xdebug](#)

[настройка в phpstorm](#)

[php основы](#)

[постман и xdebug](#)

?XDEBUG_SESSION_START=PHPSTORM

[для обработки request body](#)

[возможности репозитория](#)

[создание relation](#)

[создание внешнего ключа](#)

[создание сервиса](#)

[описание столбцов для полей сущности](#)

[трансформируем json в свой класс
валидация](#)

[сохранение в бд](#)

[ошибка: сервер уже запущен](#)

Терминальные команды

php bin/console make:entity - создание сущности

php bin/console make:controller - создание контроллера

php bin/console make:migration - создание миграции

php bin/console doctrine:migrations:migrate - выполнение всех миграций

php bin/console doctrine:migrations:execute <название миграции> --up (--down) - выполнение конкретной миграции

symfony server:start

symfony server:stop - запуск и прерывание сервера

docker-compose up

docker-compose down - запуск и прерывание контейнера в Docker

Алгоритм выполнения задания

1. С помощью docker-compose файла подняла в Docker базу данных (пустую).
2. Проанализировала файл «таблицы», чтобы определить, какие сущности мне нужны, какие между ними будут отношения.
3. Создала все необходимые сущности с помощью терминальной команды.
4. Создала миграцию по имеющимся сущностям. Немного ее исправила, убрав ненужные ключи. Выполнила миграцию. Заполнила базу данных (данные прикреплены ниже в этом файле)
5. Проанализировала основной файл, чтобы понять, какие функции мне нужны.
6. Далее для каждой сущности создавала контроллер и сервис, и прописывала нужный функционал. Сразу тестировала в postman.

Данные для базы данных*

```
INSERT INTO T001 (burks, butxt) VALUES  
( 'ABCD', 'Item 1'),  
( 'EFGH', 'Item 2'),  
( 'IJKL', 'Item 3'),  
( 'MNOP', 'Item 4'),  
( 'QRST', 'Item 5');
```

```
INSERT INTO ZTINMM_TK_H (burks_id, konkurs_id, konkurs_nr,  
konkurs_name, org_key, stat, crt_date, crt_time, crt_user) VALUES  
( 'ABCD', '123456789012', 'Konkurs 1', 'Konkurs Name 1', 'ORG', 'STAT1',  
'2021-01-01', '12:00:00', 'User1'),  
( 'EFGH', '987654321098', 'Konkurs 2', 'Konkurs Name 2', 'ORG', 'STAT2',  
'2021-01-02', '14:30:00', 'User2'),  
( 'IJKL', '112233445566', 'Konkurs 3', 'Konkurs Name 3', 'ORG', 'STAT3',  
'2021-01-03', '09:45:00', 'User3'),  
( 'MNOP', '789012345678', 'Konkurs 4', 'Konkurs Name 4', 'ORG', 'STAT4',  
'2021-01-04', '18:20:00', 'User4'),  
( 'QRST', '654321098765', 'Konkurs 5', 'Konkurs Name 5', 'ORG', 'STAT5',  
'2021-01-05', '11:10:00', 'User5');
```

```
INSERT INTO ZINMM_SOF_LOT_H (konkurs_id_id, lot_id, lot_nr,  
lot_name, ekorg, finan_limit_wvat, finan_limit_wovat, vat_rate,  
cals_nds) VALUES  
( '123456789012', 'Lot 1', 'Lot Number 1', 'Lot Name 1', 'EKOR', 1000.00,  
900.00, '20', 'Y'),  
( '987654321098', 'Lot 2', 'Lot Number 2', 'Lot Name 2', 'EKOR', 2000.00,  
1800.00, '20', 'Y'),  
( '112233445566', 'Lot 3', 'Lot Number 3', 'Lot Name 3', 'EKOR', 1500.00,  
1350.00, '20', 'Y'),  
( '789012345678', 'Lot 4', 'Lot Number 4', 'Lot Name 4', 'EKOR', 3000.00,  
2700.00, '20', 'Y'),  
( '654321098765', 'Lot 5', 'Lot Number 5', 'Lot Name 5', 'EKOR', 2500.00,  
2250.00, '20', 'Y');
```

```
INSERT INTO ZTINMM_TK_OFR (konkurs_id_id, lot_id_id, tabix, lifnr,  
lifnr_name, orf_date, orf_time, price_nds, price_with_nds, deliver_date,  
deliver_time, win_flg) VALUES
```

```
(
'123456789012', 'Lot 1', 1.23, 'LIFNR1', 'Supplier 1', '2021-01-01',
'10:00:00', 100.50, 120.60, '2021-01-05', '15:00:00', 'Y'),
('987654321098', 'Lot 2', 2.34, 'LIFNR2', 'Supplier 2', '2021-01-02',
'13:30:00', 200.75, 240.90, '2021-01-06', '18:30:00', 'N'),
('112233445566', 'Lot 3', 3.45, 'LIFNR3', 'Supplier 3', '2021-01-03',
'08:45:00', 150.25, 180.30, '2021-01-07', '11:45:00', 'Y'),
('789012345678', 'Lot 4', 4.56, 'LIFNR4', 'Supplier 4', '2021-01-04',
'17:20:00', 300.80, 361.00, '2021-01-08', '20:20:00', NULL),
('654321098765', 'Lot 5', 5.67, 'LIFNR5', 'Supplier 5', '2021-01-05',
'10:10:00', 250.50, 300.60, '2021-01-09', '12:10:00', NULL);
```

```
INSERT INTO ZTINMM_TK_PERS (konkurs_id_id, tabix, pers_func,
pers_id, pers_fio) VALUES
('123456789012', 1, 'FUNC1', '123456789012', 'John Doe'),
('987654321098', 2, 'FUNC2', '987654321098', 'Jane Smith'),
('112233445566', 3, 'FUNC3', '112233445566', 'Mike Johnson'),
('789012345678', 4, 'FUNC4', '789012345678', 'Emily Davis'),
('654321098765', 5, 'FUNC5', '654321098765', 'Daniel Wilson');
```

```
INSERT INTO ZTINMM_TK_VOTE (konkurs_id_id, lifnr_id, stat, pers_id,
vote_res, vote_finish, vote_win, vote_date, vote_time, vote_user)
VALUES
('123456789012', 'LIFNR1', 'STAT1', '123456789012', 'Y', 'Y', 'Y', '2021-01-
05', '14:00:00', 'User1'),
('987654321098', 'LIFNR2', 'STAT2', '987654321098', 'N', 'N', 'N', '2021-
01-06', '16:00:00', 'User2'),
('112233445566', 'LIFNR3', 'STAT3', '112233445566', 'Y', 'Y', 'Y', '2021-01-
07', '10:00:00', 'User3'),
('789012345678', 'LIFNR4', 'STAT4', '789012345678', NULL, NULL, NULL,
NULL, NULL, 'User4'),
('654321098765', 'LIFNR5', 'STAT5', '654321098765', NULL, NULL, NULL,
NULL, NULL, 'User5');
```

(*автоматически сгенерированы на специальном сайте по sql
выражению для создания таблиц)