

# WEB-APPLICATION ORDER YOUR DINNER

Voronov Sergey

Пожалуйста, войдите:

Username

Password

Login

Добро пожаловать, VoronovSS!

Набор Салат + Суп + Горячее + Напиток

Посмотреть супы дня

Салат С огурцом, редисом и яйцом (соус винегр

Горячее Пельмени из баранины со сметаной

Хоспер

Гарнир Без гарнира

Напиток Компот из сухофруктов

Хлеб Булочка с отрубями

Итог: 275

Баланс: 200

Подтвердить

☐ Переносить последний сделанный заказ на следующий день

logout

Текущий заказ:

Набор: Салат + Суп + Горячее + Напиток

Суп: +

Салат: С огурцом, редисом и яйцом (соус винегрет)

Горячее: Пельмени из баранины со сметаной

Хоспер:

Гарнир: Без гарнира

Напиток: Компот из сухофруктов

Хлеб: Булочка с отрубями

Удалить

<input type="checkbox"/>	Name	Date	Set	Soup	Salad	Hot	Hosper	Fixings	Bread	Drink	Sum
<input type="checkbox"/>	YahontovN	Mon Feb 19 13:53:25 MSK 2018	Обед месяца с супом	+				Без гарнира	Двойной багет	Морс брусничный	-390

Работа с записями:

Удалить выбранные

Количество раз:

0

Повторить выбранные

Имя:

Баланс:

Внести сумму:

Внести

Дата:

 2/19/18

Сумма за день: 390

Вывести таблицу

Рассылка таблиц

Email

vadim@acs-service.ru

info@kitchenpolly.ru

evgeniya.lushnikova@msk.nfretail.ru

Удалить

Добавить

☒ Включить автоотправку

Сформировать и отправить по почте

# Plain html in output file

USLY wrong with this  
s picture... trying to stuff  
inside a println()?? That  
e right...

This is how you create a dynamic web  
page in a servlet. You have to print  
the whole thing to an output stream  
(it's really part of the HTTP response  
stream that you're printing to).

```
out.println("<html> " +  
            "<body>" +  
            "<h1>Skyler\'s Login Page</h1>" +  
            "<br>" + today +  
            "</body>" +  
            "</html>");
```

```
hmap.put("\\$\\{notification}", notification);  
hmap.put("\\$\\{action}", action);  
hmap.put("\\$\\{registrationLink}", registrationLink);  
hmap.put("\\$\\{loginLink}", loginLink);  
  
Utilities.printResponse(response, htmlPath: "html/index.html", hmap);
```

TODO: reflection to work with var names

# AJAX, jQuery

Hello, user@mail.ru

Набор: Салат + Суп + Горячее

Салат: С огурцом, редисом и яйцом (соус винегрет)

Горячее: Бестроганоф из говядины с грибами

Гарнир: Рис

Напиток: Компот из сухофруктов

Итог:  
300

Баланс:  
5

Отправить

Logout

Hello, user@mail.ru

Набор: Салат + Горячее

Салат: С огурцом, редисом и яйцом (соус винегрет)

Горячее: Бестроганоф из говядины с грибами

Гарнир: Рис

Напиток: Компот из сухофруктов

Итог:  
250

Баланс:  
5

Отправить

Logout

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
<script>
    $(document).ready(function () {
        $("#set").change(function () {
            $.ajax({
                url: "SumAjax.do",
                type: "POST",
                data: ({set: $("#set").val()}),
                success: function (result) {
                    $("#sum").html(result);
                }
            });
        });
    });
</script>
```

```
try {
    setEntity = genericDaoSetEntity.getByParameters(new Field[]{SetEntity.class.getDeclaredField( name: "name")},
        new String[]{request.getParameter( s: "set")});
} catch (NoSuchFieldException e) {
    e.printStackTrace();
}
response.setContentType("text/plain");
PrintWriter out = response.getWriter();
out.print(String.valueOf(setEntity.get(0).getPrice()));
```

# JdbcTemplate

## First version:

```
public ArrayList<ArrayList<Object>> query(String sql, String[] strings){
```

```
public int queryForInt(String sql, String[] strings){
```

## Second version:

```
public <T> T selectWithMethod(String sql, String[] strings, ResultSetHandler<T> resultSetHandler) {
```

```
public interface ResultSetHandler<T> {  
    T func(ResultSet resultSet) throws SQLException;  
}
```

# DAO

```
public interface UserDao {  
    public ArrayList<UserEntity> getByRole(int roleId);  
    public UserEntity getByLogin(String login);  
    public UserEntity getByLoginPassword(String login, String password);  
    public void deleteByLogin(String login);  
    public UserEntity create(String login, String password, String role);  
    public UserEntity update(String login, String password, String role);  
}
```

```
ResultSetHandler<UserEntity> rsHandlerGetUserEntity = (resultSet) -> {  
    UserEntity userEntity = null;  
    if (resultSet.next()) {  
        userEntity = new UserEntity(resultSet.getString("LOGIN"),  
                                     resultSet.getString("PASSWORD"), resultSet.getString("ROLE"));  
    }  
    return userEntity;  
};  
  
@Override  
public UserEntity getByLogin(String login) {  
    sql = "SELECT LOGIN, PASSWORD, ROLE FROM USERS WHERE LOGIN = ?";  
    UserEntity userEntity = jdbcTemplateMy2.selectWithMethod(sql, new String[]{login}, rsHandlerGetUserEntity);  
    return userEntity;  
}
```

# GenericDAO

```
@Retention(RetentionPolicy.RUNTIME)
public @interface Table {

    String name();

}
```

@ Column

@ Entity

@ Id

@ Table

```
@Entity
@Table(name = "SETS")
public class SetEntity {

    @Id
    @Column(name = "id")
    private int id;
    @Column(name = "name")
    private String name;
    @Column(name = "price")
    private int price;

    public int getId() { return id; }

    public void setId(int id) { this.id = id; }

    public String getName() { return name; }

    public void setName(String name) { this.name = name; }

    public int getPrice() { return price; }

    public void setPrice(int price) { this.price = price; }

    public SetEntity() {

    }

}
```



# GenericDAO

```
public GenericDao(Class<T> clazz){

    this.clazz = clazz;
    Annotation annotation = clazz.getAnnotation(Table.class);
    Table table = (Table)annotation;
    this.table = table.name();

    for (Field field : clazz.getDeclaredFields()) {
        if(field.isAnnotationPresent(Id.class)){
            idField = field;
            idField.setAccessible(true);
            annotation = field.getAnnotation(Column.class);
            Column column = (Column)annotation;
            idColumn = column.name();
        }
        if(field.isAnnotationPresent(Column.class)){
            annotation = field.getAnnotation(Column.class);
            Column column = (Column)annotation;
            fieldColumns.put(field, column.name());
        }
    }

    rsHandlerForOne = (resultSet) -> {
        T entity = null;
        if (resultSet.next()) {
            try {
                entity = clazz.newInstance();
                for(Field field: entity.getClass().getDeclaredFields()){
                    field.setAccessible(true);
                    field.set(entity, resultSet.getString(fieldColumns.get(field)));
                }
            } catch (InstantiationException e) {
                e.printStackTrace();
            } catch (IllegalAccessException e) {
                e.printStackTrace();
            }
        }
        return entity;
    };
}
```

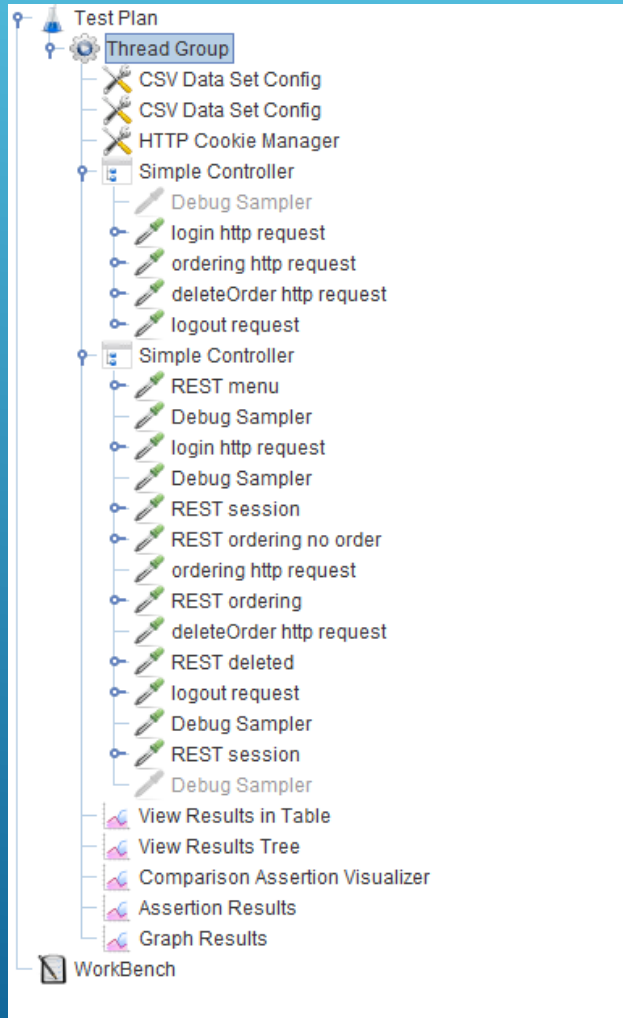
# REST API

localhost:8080/api/order?login=user@mail.ru

```
{"order":{"set": "Салат + Суп + Горячее", "salad": "С огурцом, редисом и яйцом (соус винегрет)", "hot": "Бестроганоф из говядины с грибами", "soup": "+", "fixings": "Рис", "drink": "Компот из сухофруктов", "sum": "-300"}}
```

- ▼ RESTServlets
  - Ⓢ MenuREST
  - Ⓢ OrderREST
  - Ⓢ SessionREST

# jMeter



Patterns to Test	
	Patterns to Test
1	<code>\${__escapeOroRegexpChars(\${set})}</code>
1	<code>\${salad}</code>
1	<code>\${fixings}</code>
1	<code>\${drink}</code>
1	<code>Hello, \${login}</code>

### Regular Expression Extractor

Name:

Comments:

Apply to:

☒ Main sample and sub-samples ☐ Main sample only

Field to check:

Reference Name:

Regular Expression:

Template:

Match No. (0 for Random):

Default Value:

# jMeter

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
1	15:19:49.347	Thread Group 1-1	login http request	83	✓	3999	422	11	1
2	15:19:49.401	Thread Group 1-3	login http request	61	✓	3999	422	16	1
3	15:19:49.431	Thread Group 1-1	ordering http request	44	✓	1117	590	44	0
4	15:19:49.376	Thread Group 1-2	login http request	100	✓	3995	420	20	1
5	15:19:49.463	Thread Group 1-3	ordering http request	24	✓	1117	590	24	0
6	15:19:49.477	Thread Group 1-2	ordering http request	13	✓	1116	590	13	0
7	15:19:49.434	Thread Group 1-4	login http request	127	✓	3999	422	6	1
8	15:19:49.475	Thread Group 1-1	deleteOrder http request	91	✓	3924	367	10	0
9	15:19:49.567	Thread Group 1-1	logout request	4	✓	1510	367	2	0
10	15:19:49.575	Thread Group 1-1	REST menu	21	✓	850	179	21	0
11	15:19:49.562	Thread Group 1-4	ordering http request	34	✓	1117	590	34	0
12	15:19:49.596	Thread Group 1-1	Debug Sampler	0	✓	909	0	0	0
13	15:19:49.488	Thread Group 1-3	deleteOrder http request	109	✓	3924	367	17	0
14	15:19:49.598	Thread Group 1-3	logout request	3	✓	1510	367	1	0
15	15:19:49.606	Thread Group 1-3	REST menu	17	✓	850	179	17	0
16	15:19:49.624	Thread Group 1-3	Debug Sampler	0	✓	908	0	0	0
17	15:19:49.490	Thread Group 1-2	deleteOrder http request	137	✓	3920	367	44	0
18	15:19:49.462	Thread Group 1-5	login http request	169	✓	3997	424	19	1
19	15:19:49.529	Thread Group 1-7	login http request	106	✓	3997	424	32	1
20	15:19:49.629	Thread Group 1-2	logout request	11	✓	1510	367	5	0
21	15:19:49.492	Thread Group 1-6	login http request	163	✓	3997	424	50	1
22	15:19:49.576	Thread Group 1-9	login http request	81	✓	3996	422	11	1
23	15:19:49.552	Thread Group 1-8	login http request	120	✓	3996	422	15	1
24	15:19:49.597	Thread Group 1-4	deleteOrder http request	83	✓	3924	367	19	0
25	15:19:49.638	Thread Group 1-7	ordering http request	48	✓	1118	590	48	0
26	15:19:49.681	Thread Group 1-4	logout request	7	✓	1510	367	3	0
27	15:19:49.607	Thread Group 1-10	login http request	94	✓	3996	422	4	1
28	15:19:49.658	Thread Group 1-9	ordering http request	45	✓	1117	590	45	0
29	15:19:49.624	Thread Group 1-3	login http request	97	✓	3924	475	32	0
30	15:19:49.721	Thread Group 1-3	Debug Sampler	1	✓	846	0	0	0
31	15:19:49.722	Thread Group 1-3	REST session	1	✓	165	202	1	0
32	15:19:49.724	Thread Group 1-3	REST ordering no order	12	✓	146	200	12	0
33	15:19:49.643	Thread Group 1-2	REST menu	95	✓	850	179	95	0
34	15:19:49.739	Thread Group 1-2	Debug Sampler	0	✓	906	0	0	0
35	15:19:49.673	Thread Group 1-8	ordering http request	121	✓	1117	590	121	0
36	15:19:49.656	Thread Group 1-6	ordering http request	154	✓	1118	590	154	0
37	15:19:49.684	Thread Group 1-13	login http request	127	✓	3996	422	19	2
38	15:19:49.597	Thread Group 1-1	login http request	216	✓	3924	475	16	0
39	15:19:49.632	Thread Group 1-11	login http request	181	✓	3996	422	40	1
40	15:19:49.814	Thread Group 1-1	Debug Sampler	0	✓	846	0	0	0
41	15:19:49.814	Thread Group 1-1	REST session	1	✓	165	202	1	0



**Username**

user@mail.ru

**Password**

....

Login

[Registration](#)

Hello, user@mail.ru

You have made your order

Набор: Салат + Суп + Горячее

Суп: +

Салат: С огурцом, редисом и яйцом (соус винегрет)

Горячее: Бестроганоф из говядины с грибами

Гарнир: Рис

Напиток: Компот из сухофруктов

Сумма: -300

Delete

Logout