СОГЛАСОВАНО	**УТВЕРЖДАЮ**
Проректор по научной работе РГСУ	Генеральный директор ООО "Ойстерс"
_____________А.Н. Малолетко	_______________ Н.Я. Небжицкая
« » 2017 г.	« » 2017 г.

Программно-аппаратный комплекс оператора фискальных данных
Текст программы

МОДУЛЬ ХРАНЕНИЯ ДАННЫХ

RU.63776339.04.13-01 12 01

Директор Научно-исследовательский институт перспективных направлений и технологий
_______________ М.В.Виноградова

«   »   2017 г.

Москва 2017

СОДЕРЖАНИЕ

АННОТАЦИЯ

Данный документ содержит текст программы модуль хранения данных

1 ИСХОДНЫЙ ТЕКСТ. МОДУЛЬ ХРАНЕНИЯ ДАННЫХ

```
MvcConfig.java

package ru.bema.config;

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.ComponentScan;
import org.springframework.context.annotation.Configuration;
import org.springframework.context.support.ReloadableResourceBundleMessageSource;
import org.springframework.core.Ordered;
import org.springframework.web.multipart.commons.CommonsMultipartResolver;
import org.springframework.web.servlet.config.annotation.*;
import org.springframework.web.servlet.view.JstlView;
import org.springframework.web.servlet.view.UrlBasedViewResolver;

@EnableWebMvc
@Configuration
@ComponentScan(basePackages = { "ru.bema" })
public class MvcConfig extends WebMvcConfigurerAdapter {

    @Override
    public void addResourceHandlers(ResourceHandlerRegistry registry) {
        registry.addResourceHandler("/resources/**").addResourceLocations("/resources/");
registry.addResourceHandler("/resources/script/**").addResourceLocations("/resources/script/");
registry.addResourceHandler("/resources/css/**").addResourceLocations("/resources/css/");
registry.addResourceHandler("/webjars/**").addResourceLocations("classpath:/META-INF/resources/webjars/");
    }

    @Override
    public void configureDefaultServletHandling(DefaultServletHandlerConfigurer configurer) {
        configurer.enable();
    }

    @Bean
    public UrlBasedViewResolver jspViewResolver() {
        UrlBasedViewResolver viewResolver = new UrlBasedViewResolver();
        viewResolver.setViewClass(JstlView.class);
        viewResolver.setPrefix("/WEB-INF/views/");
        viewResolver.setSuffix(".jsp");
        return viewResolver;
    }

    @Bean
    public CommonsMultipartResolver multipartResolver(){
        CommonsMultipartResolver multipartResolver = new CommonsMultipartResolver();
        multipartResolver.setMaxUploadSize(500000);
        return multipartResolver;
    }

    @Bean(name = "messageSource")
    public ReloadableResourceBundleMessageSource getMessageSource() {
        ReloadableResourceBundleMessageSource resource = new ReloadableResourceBundleMessageSource();
        resource.setBasename("classpath:messages");
        resource.setDefaultEncoding("UTF-8");
        return resource;
    }

    @Override
    public void addViewControllers(ViewControllerRegistry registry) {
        registry.addViewController("/login").setViewName("login");
        registry.setOrder(Ordered.HIGHEST_PRECEDENCE);
    }



}
-------------------------------------------------------------------------------------

LoginController.java

package ru.bema.controller;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.servlet.ModelAndView;
import ru.bema.service.dto.RegisterDto;

@Controller
public class LoginController {

    @RequestMapping(value = {"/", "/welcome**"}, method = RequestMethod.GET)
    public ModelAndView welcomePage() {

            ModelAndView model = new ModelAndView();
            model.addObject("title", "заголовок с бэкэнда 1");
            model.addObject("message", "This is welcome page!");
```

```
•           model.setViewName("home");
•           return model;
•
•       }
•
•       @RequestMapping(value = "/admin**", method = RequestMethod.GET)
•       public ModelAndView adminPage() {
•
•           ModelAndView model = new ModelAndView();
•           model.addObject("title", "заголовок с бэкэнда 2");
•           model.addObject("message", "заготовка закрытой аутонтификацией страницы");
•           model.setViewName("admin");
•
•           return model;
•
•       }
•
•       //Spring Security see this :
•       @RequestMapping(value = "/login", method = RequestMethod.GET)
•       public ModelAndView login(
•               @RequestParam(value = "error", required = false) String error,
•               @RequestParam(value = "logout", required = false) String logout) {
•
•           ModelAndView model = new ModelAndView();
•           if (error != null) {
•               model.addObject("error", "Invalid username and password!");
•           }
•
•           if (logout != null) {
•               model.addObject("msg", "You've been logged out successfully.");
•           }
•           model.setViewName("login");
•
•           return model;
•
•       }
•
•       @RequestMapping(value = "/register", method = RequestMethod.GET)
•       public ModelAndView register(
•               @RequestParam(value = "error", required = false) String error,
•               @RequestParam(value = "logout", required = false) String logout) {
•
•           ModelAndView model = new ModelAndView();
•           if (error != null) {
•               model.addObject("error", "Invalid username and password!");
•           }
•
```

```
                if (logout != null) {
                        model.addObject("msg", "You've been logged out successfully.");
                }
                model.setViewName("register");

                return model;

        }

        @RequestMapping(value = "/doRegister", method = RequestMethod.POST)
        public ModelAndView doRegister(
                @ModelAttribute RegisterDto registerFormData,
                @RequestParam(value = "error", required = false) String error,
                @RequestParam(value = "logout", required = false) String logout
        ) {
            return welcomePage();

        }
    }

    --------------------------------------------------------------------------------

    User.java

    package ru.bema.persist.entity;


    import org.hibernate.annotations.GenericGenerator;

    import javax.persistence.*;
    import java.util.Date;
    import java.util.HashSet;
    import java.util.Set;

    @Entity
    @Table(name = "users")
    public class User {

        @Id
        @GenericGenerator(name = "generator", strategy = "increment")
        @GeneratedValue(generator = "generator")
        @Column(name = "id", nullable = false)
        private Long id;

        @Column(name = "first_name", nullable = false)
        private String firstName;

        @Column(name = "family_name", nullable = false)
```

```
	private String familyName;

	@Column(name = "e_mail", nullable = false)
	private String email;

	@Column(name = "phone", nullable = false)
	private String phone;

	@Column(name = "language", nullable = false)
	private String language;

	@Column(name = "login", nullable = false)
	private String login;

	@Column(name = "password", nullable = false)
	private String password;

	@Column(name = "birth_date")
	private Date birthDate;

	@Column(name = "enabled")
	private Boolean enabled;

	@ManyToMany(fetch = FetchType.EAGER)
	@JoinTable(name = "users_authority",
			joinColumns = {
				@JoinColumn(name = "id_user", referencedColumnName = "id")
			},
			inverseJoinColumns = {
				@JoinColumn(name = "id_authority", table = "authority", referencedColumnName = "id")
			}
	)
	private Set<AuthorityDto> authorities = new HashSet<>();

	public String getFirstName() {
		return firstName;
	}

	public void setFirstName(String firstName) {
		this.firstName = firstName;
	}

	public String getFamilyName() {
		return familyName;
	}

```

```
•	public void setFamilyName(String familyName) {
•		this.familyName = familyName;
•	}
•
•	public long getId() {
•		return id;
•	}
•
•	public void setId(long id) {
•		this.id = id;
•	}
•
•	public String getLogin() {
•		return login;
•	}
•
•	public void setLogin(String login) {
•		this.login = login;
•	}
•
•	public String getPassword() {
•		return password;
•	}
•
•	public void setPassword(String password) {
•		this.password = password;
•	}
•
•	public Date getBirthDate() {
•		return birthDate;
•	}
•
•	public void setBirthDate(Date birthDate) {
•		this.birthDate = birthDate;
•	}
•
•	public Set<AuthorityDto> getAuthorities() {
•		return authorities;
•	}
•
•	public void setAuthorities(Set<AuthorityDto> authorities) {
•		this.authorities = authorities;
•	}
•
•	public Boolean getEnabled() {
•		return enabled;
•	}
•
```

```
    public void setEnabled(Boolean enabled) {
        this.enabled = enabled;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    public String getLanguage() {
        return language;
    }

    public void setLanguage(String language) {
        this.language = language;
    }

    public String getPhone() {
        return phone;
    }

    public void setPhone(String phone) {
        this.phone = phone;
    }

}

--------------------------------------------------------------------------------------------


RegisterDto.java

package ru.bema.service.dto;

import java.io.Serializable;

/**
 * Created by pixie on 21.04.2017.
 */
public class RegisterDto implements Serializable {
   private String username;
   private String password;
   private String email;
   private String firstname;
   private String secondname;
```

```
•           private String lastname;
•
•           public String getUsername() {
•               return username;
•           }
•
•           public void setUsername(String username) {
•               this.username = username;
•           }
•
•           public String getPassword() {
•               return password;
•           }
•
•           public void setPassword(String password) {
•               this.password = password;
•           }
•
•           public String getEmail() {
•               return email;
•           }
•
•           public void setEmail(String email) {
•               this.email = email;
•           }
•
•           public String getFirstname() {
•               return firstname;
•           }

•           public void setFirstname(String firstname) {
•               this.firstname = firstname;
•           }
•
•           public String getSecondname() {
•               return secondname;
•           }
•
•           public void setSecondname(String secondname) {
•               this.secondname = secondname;
•           }
•
•           public String getLastname() {
•               return lastname;
•           }
•
•           public void setLastname(String lastname) {
•               this.lastname = lastname;
•           }
•       }
•
•
•       -------------------------------------------------------------------------------------------
•
•       UserDto.java
•
•       package ru.bema.service.dto;
•
•       import org.apache.commons.lang3.StringUtils;
•       import org.dozer.Mapping;
•
•       import java.util.Date;
import java.util.HashSet;
import java.util.Set;

public class UserDto {

    @Mapping("id")
    private Long id;

    @Mapping("firstName")
    private String firstName;

    @Mapping("familyName")
    private String familyName;

    @Mapping("email")
    private String email;

    @Mapping("phone")
    private String phone;

    @Mapping("language")
    private String language;

    @Mapping("login")
    private String login;

    @Mapping("password")
    private String password;

    @Mapping("burthDate")
    private Date burthDate;

    @Mapping("authorities")
    private Set<AuthorityDto> authorities = new HashSet<>();

    @Mapping("enabled")
    private Boolean enabled;

    @Mapping("pictureId")
    private Long pictureId;

    public Long getId() {
        return id;
    }

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

    public String getFirstName() {
        return firstName;
    }

    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }

    public String getFamilyName() {
        return familyName;
    }

    public void setFamilyName(String familyName) {
        this.familyName = familyName;
    }

    public String getLogin() {
        return login;
    }

    public void setLogin(String login) {
        this.login = login;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }

    public Date getBurthDate() {
        return burthDate;
    }

    public void setBurthDate(Date burthDate) {
        this.burthDate = burthDate;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    public String getPhone() {
        return phone;
    }

    public void setPhone(String phone) {
        this.phone = phone;
    }

    public String getLanguage() {
        return language;
    }

    public void setLanguage(String language) {
        this.language = language;
    }

    public Set<AuthorityDto> getAuthorities() {
        return authorities;
    }

    public void setAuthorities(Set<AuthorityDto> authorities) {
        this.authorities = authorities;
    }

    public Boolean getEnabled() {
        return enabled;
    }

    public void setEnabled(Boolean enabled) {
        this.enabled = enabled;
    }

    public Long getPictureId() {
        return pictureId;
    }

    public void setPictureId(Long pictureId) {
        this.pictureId = pictureId;
    }

    public String getAuthoritiesAsString() {
        StringBuilder sb = new StringBuilder();
        for (AuthorityDto a : this.getAuthorities()) {
            sb.append(a.getName());
            sb.append(", ");
        }
        return StringUtils.substring(sb.toString(), 0, sb.length() - 2);
    }

}
```