ID	Original Documentation A	Documentation - LLM B	Documentation - LLM C	Documentation - LLM D
1	Original code without LLM modification	Code with LLM change	Code with LLM change	Code with LLM change
1	// Arquivo: AWSConfig.java	/**	/**	/**
'	// Caminho: \br\gov\ana\sarsb\config\AWSConfig.java	* Creates AWS credentials for S3 access.	* Creates AWS credentials for S3 access.	* Cria credenciais AWS para acesso ao S3. * ATENÇÃO: As credenciais são
	package br.gov.ana.sarsb.config;	* @return AWS credentials with access key and secret key	* @return AWS credentials with access key and secret key	carregadas de configuração externa * por questões de segurança.
	import com.amazonaws.auth.AWSCredentials;	*/	*/	*
	import	public AWSCredentials credentials() {	public AWSCredentials credentials() {	* @return credenciais AWS com chave de acesso e
	com.amazonaws.auth.AWSStaticCredentialsProvider; import com.amazonaws.auth.BasicAWSCredentials;	AWSCredentials credentials = new BasicAWSCredentials(AWSCredentials credentials = new BasicAWSCredentials(chave secreta * @since 1.0
	import com.amazonaws.regions.Regions;	" credencial ocultada pela pesquisa",	" credencial ocultada pela pesquisa ",	*/
	import com.amazonaws.services.s3.AmazonS3;	" credencial ocultada pela pesquisa "	" credencial ocultada pela pesquisa "	public AWSCredentials credentials() {
	import););	// NOTA: Credenciais devem ser carregadas de
	com.amazonaws.services.s3.AmazonS3ClientBuilder;	return credentials;	return credentials;	variáveis de ambiente ou AWS Secrets Manager
	import org.springframework.context.annotation.Bean; import	}	}	AWSCredentials credentials = new BasicAWSCredentials(
	org.springframework.context.annotation.Configuration;	/**	/**	System.getenv("AWS_ACCESS_KEY_ID"),
		* Creates and configures an Amazon S3 client	* Creates and configures an Amazon S3 client	System.getenv("AWS_SECRET_ACCESS_KEY")
	@Configuration	bean.	bean.);
	public class AWSConfig {	*	*	return credentials;
	public AWSCredentials credentials() {	* @return configured AmazonS3 client for SA_EAST_1 region	* @return configured AmazonS3 client for SA_EAST_1 region	}
	AWSCredentials credentials = new	*/	*/	/**
	BasicAWSCredentials(@Bean	@Bean	* Cria e configura um bean cliente Amazon S3.
	"credencial ocultada pela pesquisa",	public AmazonS3 amazonS3() {	public AmazonS3 amazonS3() {	*
	" credencial ocultada pela pesquisa "	AmazonS3 s3client = AmazonS3ClientBuilder .standard()	AmazonS3 s3client = AmazonS3ClientBuilder .standard()	* @return cliente AmazonS3 configurado para a região SA_EAST_1 (São Paulo)
	return credentials;	.withCredentials(new	.withCredentials(new	* @since 1.0
	}	AWSStaticCredentialsProvider(credentials()))	AWSStaticCredentialsProvider(credentials()))	*/
		.withRegion(Regions.SA_EAST_1)	.withRegion(Regions.SA_EAST_1)	@Bean
	@Bean	.build();	.build();	public AmazonS3 amazonS3() {
	public AmazonS3 amazonS3() { AmazonS3 s3client = AmazonS3ClientBuilder	return s3client;	return s3client;	AmazonS3 s3client = AmazonS3ClientBuilder .standard()
	.standard()	,	,	.withCredentials(new
	.withCredentials(new			AWSStaticCredentialsProvider(credentials()))
	AWSStaticCredentialsProvider(credentials()))			.withRegion(Regions.SA_EAST_1)
	.withRegion(Regions.SA_EAST_1) .build();			.build(); return s3client;
	return s3client;			}
	}			,
	}			
	// Fim do arquivo: AWSConfig.java			

ID	Original Documentation A	Documentation - LLM B	Documentation - LLM C	Documentation - LLM D
	Original code without LLM modification	Code with LLM change	Code with LLM change	Code with LLM change
2	// Arquivo: CorsConfiguration.java	/**	/**	/**
_	// Caminho: \br\gov\ana\sarsb\config\CorsConfiguration.java	* Configures CORS (Cross-Origin	* [Melhoria Amazon Q + ChatGPT]	* Configuração de CORS (Cross-Origin
		Resource Sharing) settings for the	* Este pacote foi revisado e expandido	Resource Sharing) para a aplicação.
	package br.gov.ana.sarsb.config;	application.	automaticamente para garantir maior completude (ΔC),	* Define políticas de acesso cross-origin para
		*	clareza (ΔH) e precisão factual (ΔT).	APIs REST.
	import org.springframework.context.annotation.Bean;	* @return WebMvcConfigurer with	* As descrições foram enriquecidas com contexto de	*
	import org.springframework.context.annotation.Configuration;	CORS configuration	uso, dependências e finalidade arquitetural.	* @author Sistema SARSB
	import org.springframework.web.servlet.config.annotation.CorsRegistry;	*/	*/	* @version 1.0
	import	@Bean		* @since 1.0
	org.springframework.web.servlet.config.annotation.WebMvcConfigurer;	public WebMvcConfigurer		*/
		corsConfigurer() {		
	@Configuration	return new WebMvcConfigurer() {	/**	
	public class CorsConfiguration {	@Override	* Configures CORS (Cross-Origin Resource Sharing)	/**
		public void	settings for the application.	* Configura as definições de CORS para a
	@Bean	addCorsMappings(CorsRegistry	*	aplicação.
	<pre>public WebMvcConfigurer corsConfigurer() {</pre>	registry) {	* @return WebMvcConfigurer with CORS configuration	* Permite acesso de origens específicas com
	return new WebMvcConfigurer() {	registry.addMapping("/**")	*/	métodos HTTP definidos.
	@Override	.allowedOrigins(@Bean	*
	public void addCorsMappings(CorsRegistry		public WebMvcConfigurer corsConfigurer() {	* @return configurador WebMvc com
	registry) {	"https://www.ana.gov.br",	return new WebMvcConfigurer() {	definições de CORS
	registry.addMapping("/**")		@Override	* @since 1.0
		"https://www.snirh.gov.br",	public void addCorsMappings(CorsRegistry registry) {	*/
	.allowedOrigins(registry.addMapping("/**")	@Bean
		"https://dev.snirh.gov.br",	.allowedOrigins(public WebMvcConfigurer corsConfigurer() {
	"https://www.ana.gov.br",		"https://www.ana.gov.br",	return new WebMvcConfigurer() {
	"https://www.snirh.gov.br",	"https://hmg.snirh.gov.br",	"https://www.snirh.gov.br",	@Override
	"https://dev.snirh.gov.br",		"https://dev.snirh.gov.br",	public void addCorsMappings(CorsRegistry
	"https://hmg.snirh.gov.br",	"http://desenvolvimento.snirh.gov.br",	"https://hmg.snirh.gov.br",	registry) {
	"http://desenvolvimento.snirh.gov.br",	"http://localhost:4200",	"http://desenvolvimento.snirh.gov.br",	registry.addMapping("/**")
	"http://localhost:4200",	"http://localhost:8080",	"http://localhost:4200",	.allowedOrigins(
	"http://localhost:8080",	Weater Males and a side of the West	"http://localhost:8080",	"https://www.ana.gov.br",
	"http://desenv.snirh.gov.br",	"http://desenv.snirh.gov.br",	"http://desenv.snirh.gov.br",	"https://www.snirh.gov.br",
	"http://desenv.snirh.gov.br/sasb/api/",	"http://docopy.opirh.gov.br/ooob/opi/"	"http://desenv.snirh.gov.br/sasb/api/",	"https://dev.snirh.gov.br",
	"https://desenv.snirh.gov.br",	"http://desenv.snirh.gov.br/sasb/api/",	"https://desenv.snirh.gov.br",	"https://hmg.snirh.gov.br",
	"https://decemy.opirh.gov.hr/coch/opi/"	"https://docopy.opirh.gov.hr"	"https://desenv.snirh.gov.br/sasb/api/"	"http://desenvolvimento.snirh.gov.br",
	"https://desenv.snirh.gov.br/sasb/api/"	"https://desenv.snirh.gov.br",	, allowedMethods("GET", "POST", "PUT",	"http://localhost:4200", "http://localhost:8080",
	1	"https://desenv.snirh.gov.br/sasb/api/"	"DELETE", "OPTIONS")	"http://desenv.snirh.gov.br",
	.allowedMethods("GET", "POST", "PUT", "DELETE", "OPTIONS")	πτιρο.//ueseriv.siii(11.guv.bi/sasb/aμi/	.allowedHeaders("Authorization", "Content-	"http://desenv.snirh.gov.br/sasb/api/",
	.autowedinethous (GET, FOST, FOT, DELETE, OFHONS)	, .allowedMethods("GET",	Type", "Accept")	"https://desenv.snirh.gov.br",
	.allowedHeaders("Authorization", "Content-Type", "Accept")	"POST", "PUT", "DELETE", "OPTIONS")	.allowCredentials(true)	"https://desenv.snirh.gov.br/sasb/api/"
	.allowedi loudel of Addition Content-Type, Accept)	. SSI, ISI, BELLIE, SI HONS)	.maxAge(3600);)
	.allowCredentials(true)	.allowedHeaders("Authorization",	}	.allowedMethods("GET", "POST", "PUT",
	.a.a Groupinato(trao)	"Content-Type", "Accept")	};	"DELETE", "OPTIONS")
	.maxAge(3600);	.allowCredentials(true)	\	.allowedHeaders("Authorization",
	}	.maxAge(3600);		"Content-Type", "Accept")
	};	}		.allowCredentials(true)
	} ″	};		.maxAge(3600);
	}	}		}
	// Fim do arquivo: CorsConfiguration.java	-		};
	· · · · · · · · · · · · · · · · · · ·]}

ID	Original Documentation A	Documentation - LLM B	Documentation - LLM C	Documentation - LLM D
טו	_			
	Original code without LLM modification	Code with LLM change	Code with LLM change	Code with LLM change
3	// Arquivo: ConfigCache.java	/**	/**	/**
	// Caminho:	* Returns the singleton instance of ConfigCache.	* [Melhoria Amazon Q + ChatGPT]	* Cache singleton para configurações da aplicação.
	\br\gov\ana\sarsb\config\cache\ConfigCache.java	*	* Este pacote foi revisado e expandido automaticamente	* Implementa cache em memória com fallback para fonte de configuração
	package br.gov.ana.sarsb.config.cache;	* @return the singleton ConfigCache instance	para garantir maior completude (ΔC), clareza (ΔH) e	externa.
	import java.io.Serializable;	*/	precisão factual (ΔT).	*
	import java.util.HashMap;	public static ConfigCache getInstance() {	* As descrições foram enriquecidas com contexto de	* @author Sistema SARSB
	public class ConfigCache implements Serializable {	if(instance == null) {	uso, dependências e finalidade arquitetural.	* @version 1.0
	private static final long serialVersionUID = -	instance = new ConfigCache();	*/	* @since 1.0
	4806648601087890853L;	l		*/
	400040001007000002,	return instance;		
	private static ConfigCooks instance:	1	/**	/**
	private static ConfigCache instance;	3	'	+ Datama a instância singleton de OsofigOsoba
	private HashMap <string, string=""> configs = new</string,>	ALC:	* Returns the singleton instance of ConfigCache.	* Retorna a instância singleton de ConfigCache.
	HashMap<>();	/**	*	*
	private transient ConfigSource source;	* Retrieves a configuration value by key.	* @return the singleton ConfigCache instance	* @return a instância singleton ConfigCache
	private ConfigCache() {	* If not found in cache, searches using the	*/	* @since 1.0
	}	configured source.	public static ConfigCache getInstance() {	*/
		*	if(instance == null) {	public static ConfigCache getInstance() {
	public static ConfigCache getInstance() {	* @param chave the configuration key	instance = new ConfigCache();	if(instance == null) {
	if(instance == null) {	* @return the configuration value or null if not found	}	synchronized(ConfigCache.class) {
	instance = new ConfigCache();	*/	return instance;	if(instance == null) {
	1	public String get(String chave) {	1	instance = new ConfigCache();
	return instance;		} /**	nistance - new connection (),
	return instance;	String valor = configs.get(chave);	'	, }
	}	if(valor == null && source != null) {	* Retrieves a configuration value by key.	}
		valor = source.pesquisar(chave);	* If not found in cache, searches using the configured	}
	public String get(String chave) {	configs.put("chave", valor);	source.	return instance;
	String valor = configs.get(chave);	}	*	}
	if(valor == null && source != null) {	return valor;	* @param chave the configuration key	/**
	valor = source.pesquisar(chave);	}	* @return the configuration value or null if not found	* Recupera um valor de configuração por chave.
	configs.put("chave", valor);		*/	* Se não encontrado no cache, busca usando a fonte configurada.
	}	/**	public String get(String chave) {	*
	return valor;	* Sets the configuration source for retrieving values.	String valor = configs.get(chave);	* @param chave a chave de configuração
	}	*	if(valor == null && source != null) {	* @return o valor de configuração ou {@code null} se não encontrado
	,	* @param source the ConfigSource to use for	valor = source.pesquisar(chave);	* @since 1.0
	nublic void actCourse(ConfigCourse course) (*/
	public void setSource(ConfigSource source) {	retrieving configuration values */	configs.put("chave", valor);	
	this.source = source;		}	public String get(String chave) {
	}	public void setSource(ConfigSource source) {	return valor;	String valor = configs.get(chave);
		this.source = source;	}	if(valor == null && source != null) {
	public void clear() {	}	/**	valor = source.pesquisar(chave);
	configs.clear();		* Sets the configuration source for retrieving values.	if(valor != null) {
	}	/**	*	configs.put(chave, valor); // Corrigido: usar 'chave' em vez de string literal
		* Clears all cached configuration values.	* @param source the ConfigSource to use for retrieving	}
	}	*/	configuration values	}
		public void clear() {	*/	return valor;
		configs.clear();	public void setSource(ConfigSource source) {	}
	// Fim do arquivo: ConfigCache.java] }	this.source = source;	, /**
	77 Tim de diquiter comigedentique	'	1	* Define a fonte de configuração para recuperar valores.
			\frac{1}{1**}	*
			* Clears all eached configuration values	* @norom course a ConfigCourse norouser no recupercaño de cele
			* Clears all cached configuration values.	* @param source a ConfigSource para usar na recuperação de valores de
			*/	configuração
			public void clear() {	*@since 1.0
			configs.clear();	*/
			}	public void setSource(ConfigSource source) {
				this.source = source;
				}
				/**
				* Limpa todos os valores de configuração em cache.
				*
				* @since 1.0
				*/ wsince 1.0 */
				public void clear() {
				configs.clear();
				}

I	Original Documentation A	Documentation - LLM B	Documentation - LLM C	Documentation - LLM D
D	Original code without LLM modification	Code with LLM change	Code with LLM change	Code with LLM change
5	// Arquivo: Http401UnauthorizedEntryPoint.java	/**	/**	/**
	// Caminho:	* Handles authentication entry point by sending HTTP	* [Melhoria Amazon Q + ChatGPT]	* Ponto de entrada para tratamento de acesso não
	\br\gov\ana\sarsb\config\security\handler\Http401UnauthorizedEntryP	401 Unauthorized response.	* Este pacote foi revisado e expandido automaticamente	autorizado.
	oint.java	*	para garantir maior completude (ΔC), clareza (ΔH) e	* Retorna resposta HTTP 401 para requisições não
		* @param request the HTTP servlet request	precisão factual (ΔT).	autenticadas.
	package br.gov.ana.sarsb.config.security.handler;	* @param response the HTTP servlet response	* As descrições foram enriquecidas com contexto de	*
		* @param authenticationException the authentication	uso, dependências e finalidade arquitetural.	* @author Sistema SARSB
	import java.io.IOException;	exception that triggered this entry point	*/	* @version 1.0
		* @throws IOException if an input or output exception		* @since 1.0
	import javax.servlet.ServletException;	occurs		*/
	import javax.servlet.http.HttpServletRequest;	* @throws ServletException if a servlet-specific		
	import javax.servlet.http.HttpServletResponse;	exception occurs	/**	
		*/	* Handles authentication entry point by sending HTTP	
	import org.springframework.security.core.AuthenticationException;	@Override	401 Unauthorized response.	/**
	import org.springframework.security.web.AuthenticationEntryPoint;	public void commence(HttpServletRequest request,	*	* Manipula ponto de entrada de autenticação enviando
	import org.springframework.stereotype.Component;	HttpServletResponse response,	* @param request the HTTP servlet request	resposta HTTP 401 Unauthorized.
		AuthenticationException authenticationException)	* @param response the HTTP servlet response	*
	@Component	throws IOException, ServletException {	* @param authenticationException the authentication	* @param request a requisição HTTP servlet
	public class Http401UnauthorizedEntryPoint implements		exception that triggered this entry point	* @param response a resposta HTTP servlet
	AuthenticationEntryPoint {		* @throws IOException if an input or output exception	* @param authenticationException a exceção de
		response.sendError(HttpServletResponse.SC_UNAUTHO	occurs	autenticação que acionou este ponto de entrada
	private static final String ACESSO_NEGADO = "Acesso Negado.";	RIZED, ACESSO_NEGADO);	* @throws ServletException if a servlet-specific	* @throws IOException se ocorrer exceção de entrada ou
		}	exception occurs	saída
	@Override		*/	* @throws ServletException se ocorrer exceção
	public void commence(HttpServletRequest request,		@Override	específica de servlet
	HttpServletResponse response,		public void commence(HttpServletRequest request,	* @since 1.0
	AuthenticationException authenticationException) throws		HttpServletResponse response,	· ·
	IOException, ServletException {		AuthenticationException authenticationException) throws IOException, ServletException {	@Override
	roonanaa aandErrar/HttpCandatDaananaa CO HNAHTHADIZED		tillows locaception, Servicetexception (public void commence(HttpServletRequest request,
	response.sendError(HttpServletResponse.SC_UNAUTHORIZED, ACESSO NEGADO);			HttpServletResponse response, AuthenticationException authenticationException)
	ACESSO_NEGADOJ;		response.sendError(HttpServletResponse.SC_UNAUTHO	throws IOException, ServletException {
	1		RIZED, ACESSO NEGADO);	tillows location, servicitate prioris
	ſ		NIZLD, AOLOGO_NEGADO),	
	ı		1	response.sendError(HttpServletResponse.SC_UNAUTHO
	ı			RIZED, ACESSO_NEGADO);
	// Fim do arquivo: Http401UnauthorizedEntryPoint.java			NIZED, AGEGGO_NEGADO),
Ь	77 Firm do disquivo. Fittip40 FornautiiorizedEntityFoint.java	1	1	1

D	Original Documentation A	Documentation - LLM B	Documentation - LLM C	Documentation - LLM D
	Original code without LLM modification	Code with LLM change	Code with LLM change	Code with LLM change
$^{+}$	// Arquivo: CustomUserDTO.java	/**	/**	/**
	// Caminho:	* Constructs a CustomUserDTO with basic	* [Melhoria Amazon Q + ChatGPT]	* DTO personalizado para detalhes do usuário
1	\br\gov\ana\sarsb\config\security\dto\CustomUserDTO.java	authentication information.	* Este pacote foi revisado e expandido	autenticado.
	ibi igoviana isaisbiconing isecunty luto ioustomoser Dio.java	*	·	
	nackaga by gay and careb config accurity dta	* @naram uparnama tha uparnama idantif ii th -	automaticamente para garantir maior completude	* Estende UserDetails do Spring Security com
	package br.gov.ana.sarsb.config.security.dto;	* @param username the username identifying the	(ΔC) , clareza (ΔH) e precisão factual (ΔT) .	informações adicionais do usuário.
	import java.util.Collection;	user whose data is represented by this UserDetails	* As descrições foram enriquecidas com contexto de	1 0 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
	import org.springframework.security.core.GrantedAuthority;	instance	uso, dependências e finalidade arquitetural.	* @author Sistema SARSB
	import org.springframework.security.core.userdetails.User;	* @param password the password used to	*/	* @version 1.0
	import lombok.EqualsAndHashCode;	authenticate the user		* @since 1.0
	import lombok.Getter;	* @param authorities the authorities granted to the		*/
	import lombok.Setter;	user		
		*/	/**	
	@EqualsAndHashCode(callSuper = false)	public CustomUserDTO(String username, String	* Constructs a CustomUserDTO with basic	
	@Getter	password, Collection extends GrantedAuthority	authentication information.	/**
	@Setter	authorities) {	*	* Constrói um CustomUserDTO com informações
	public class CustomUserDTO extends User {	, ,	* Operam username the username identifying the	*
	·	super(username, password, authorities);	* @param username the username identifying the	básicas de autenticação.
	private static final long serialVersionUID = 1L;	3	user whose data is represented by this UserDetails	+ 6
1	private String fullName;		instance	* @param username o username identificando o
	private String givenName;	/**	* @param password the password used to	usuário cujos dados são representados por esta
	private String lastName;	* Constructs a CustomUserDTO with authentication	authenticate the user	instância UserDetails
	private String title;	information and additional user details.	* @param authorities the authorities granted to the	* @param password a senha usada para autentica
	private String department;	*	user	usuário
	private String mail;	* @param username the username identifying the	*/	* @param authorities as autoridades concedidas
	/**	user whose data is represented by this UserDetails	public CustomUserDTO(String username, String	usuário
	* Constructs a CustomUserDTO with basic	instance	password, Collection extends GrantedAuthority	* @since 1.0
	authentication information.	* @param password the password used to	authorities) {	*/
	*	authenticate the user	* *	
	* @naram usarnama tha usarraidtif-inth		super(username, password, authorities);	public CustomUserDTO(String username, String
	* @param username the username identifying the user	* @param fullName the full name of the user	1	password, Collection extends GrantedAuthority</td
	whose data is represented by this UserDetails instance	* @param mail the email address of the user		authorities) {
	* @param password the password used to authenticate	* @param authorities the authorities granted to the	/**	super(username, password, authorities);
	the user	user	* Constructs a CustomUserDTO with authentication	}
	* @param authorities the authorities granted to the user	*/	information and additional user details.	
	*/	public CustomUserDTO(String username, String	*	/**
	public CustomUserDTO(String username, String	password, String fullName, String mail, Collection </td <td>* @param username the username identifying the</td> <td>* Constrói um CustomUserDTO com informações</td>	* @param username the username identifying the	* Constrói um CustomUserDTO com informações
	password, Collection extends GrantedAuthority	extends GrantedAuthority> authorities) {	user whose data is represented by this UserDetails	autenticação e detalhes adicionais do usuário.
	authorities) {	super(username, password, authorities);	instance	*
	super(username, password, authorities);	this.fullName = fullName;	* @param password the password used to	* @param username o username identificando o
	l	this.mail = mail;		1
I	} /**	uno.man - man,	authenticate the user	usuário cujos dados são representados por esta
1	•	1	* @param fullName the full name of the user	instância UserDetails
	* Constructs a CustomUserDTO with authentication		* @param mail the email address of the user	* @param password a senha usada para autentica
	information and additional user details.		* @param authorities the authorities granted to the	usuário
	*		user	* @param fullName o nome completo do usuário
	* @param username the username identifying the user		*/	* @param mail o endereço de email do usuário
	whose data is represented by this UserDetails instance		public CustomUserDTO(String username, String	* @param authorities as autoridades concedidas
	* @param password the password used to authenticate		password, String fullName, String mail, Collection </td <td>usuário</td>	usuário
	the user		extends GrantedAuthority> authorities) {	* @since 1.0
	* @param fullName the full name of the user		super(username, password, authorities);	*/
	* @param mail the email address of the user		this.fullName = fullName;	public CustomUserDTO(String username, String
	* @param authorities the authorities granted to the user		this.mail = mail;	password, String fullName, String mail, Collection
I	*/		1	extends GrantedAuthority> authorities) {
			1	
1	public CustomUserDTO(String username, String			super(username, password, authorities);
	password, String fullName, String mail, Collection </td <td></td> <td></td> <td>this.fullName = fullName;</td>			this.fullName = fullName;
I	extends GrantedAuthority> authorities) {			this.mail = mail;
	super(username, password, authorities);			}
	this.fullName = fullName;			
	this.mail = mail;			
1	}			
	}			
- 1	// Fim do arquivo: CustomUserDTO.java		1	

1	Original Documentation A	Documentation - LLM B	Documentation - LLM C	Documentation - LLM D
D.	Original code without LLM modification	Code with LLM change	Code with LLM change	Code with LLM change
	// Arquivo: AcoesDocContratoMatrizEnumConverter.java	/**	/**	/**
7	// Arquivo: AcoesDocContratoMatrizEnumConverter.java // Caminho:	* Converts enum attribute to database column value.	'	* Conversor JPA para enum
		Converts enum attribute to database column value.	* • [Melhoria Amazon Q + ChatGPT]	•
	\br\gov\ana\sarsb\model\converter\AcoesDocContratoMatrizEnumConverte	* O	* Este pacote foi revisado e expandido	AcoesDocContratoMatrizEnum.
	r.java	* @param attribute the enum attribute to convert	automaticamente para garantir maior completude (ΔC),	* Converte entre valores de enum e representação de
	and the state of t	* @return the enum code as Long, or null if attribute is	clareza (Δ H) e precisão factual (Δ T).	banco de dados.
	package br.gov.ana.sarsb.model.converter;	null */	* As descrições foram enriquecidas com contexto de	* O a a three O'at a war OAROR
	in the second se	•	uso, dependências e finalidade arquitetural.	* @author Sistema SARSB
	import br.gov.ana.sarsb.model.enums.AcoesDocContratoMatrizEnum;	@Override	*/	* @version 1.0
	the second transfer of	public Long		* @since 1.0
	import javax.persistence.AttributeConverter;	convertToDatabaseColumn(AcoesDocContratoMatrizE		*/
	import javax.persistence.Converter;	num attribute) {		
	00	if (attribute == null) {	/**	
	@Converter(autoApply = true)	return null;	* Converts enum attribute to database column value.	/**
	public class AcoesDocContratoMatrizEnumConverter implements	}	*	'
	AttributeConverter <acoesdoccontratomatrizenum, long=""> {</acoesdoccontratomatrizenum,>	return attribute.getCodigo();	* @param attribute the enum attribute to convert	* Converte atributo enum para valor de coluna do
		}	* @return the enum code as Long, or null if attribute is	banco de dados.
	@Override	/**	null	
	public Long convertToDatabaseColumn(AcoesDocContratoMatrizEnum	·	*/	* @param attribute o atributo enum para converter
	attribute) {	* Converts database column value to enum attribute.	@Override	* @return o código do enum como Long, ou {@code
	if (attribute == null) {		public Long	null} se attribute for {@code null}
	return null;	* @param dbData the database value to convert	convertToDatabaseColumn(AcoesDocContratoMatrizE	* @since 1.0
	}	* @return the corresponding enum value, or null if	num attribute) {	*/
	return attribute.getCodigo();	dbData is null	if (attribute == null) {	@Override
	}	*/	return null;	public Long
		@Override	}	convertToDatabaseColumn(AcoesDocContratoMatrizE
	@Override	public AcoesDocContratoMatrizEnum	return attribute.getCodigo();	num attribute) {
	public AcoesDocContratoMatrizEnum convertToEntityAttribute(Long	convertToEntityAttribute(Long dbData) {	}	if (attribute == null) {
	dbData) {	if (dbData == null) {		return null;
	if (dbData == null) {	return null;	/**	}
	return null;	}	* Converts database column value to enum attribute.	return attribute.getCodigo();
	}	return AcoesDocContratoMatrizEnum.get(dbData);	*	}
	return AcoesDocContratoMatrizEnum.get(dbData);	}	* @param dbData the database value to convert	
	}		* @return the corresponding enum value, or null if	/**
	}		dbData is null	* Converte valor de coluna do banco de dados para
			*/	atributo enum.
			@Override	*
	// Fim do arquivo: AcoesDocContratoMatrizEnumConverter.java		public AcoesDocContratoMatrizEnum	* @param dbData o valor do banco de dados para
			convertToEntityAttribute(Long dbData) {	converter
			if (dbData == null) {	* @return o valor enum correspondente, ou {@code
			return null;	null} se dbData for {@code null}
			}	* @since 1.0
			return AcoesDocContratoMatrizEnum.get(dbData);	*/
			}	@Override
				public AcoesDocContratoMatrizEnum
				convertToEntityAttribute(Long dbData) {
l				if (dbData == null) {
l				return null;
				}
				return AcoesDocContratoMatrizEnum.get(dbData);
				}

Documentation A Documentation					
8 / Auguste formulations/ethiology loops	ID	Original Documentation A	Documentation - LLM B	Documentation - LLM C	Documentation - LLM D
A Comment		Original code without LLM modification	Code with LLM change	Code with LLM change	Code with LLM change
A Comment	Ω	// Arquivo: FormularioAditivoDTO.java	/**	/**	/**
torque/mancestratives/independent accordance control (1) cycle public for a finite or maintained accordance of the finite or maintained accordance or maintained acco	0		* Converts a FormularioAditivoAgencia entity to	* [Melhoria Amazon O + ChatGPT]	* DTO para formulários de aditivo de agência reguladora.
percent pile ingremen semanth model durin degree semanth model durin degree semanth model in the percentage of the perce				_ ·	
import income and mode with giftermidens/deficive/genous; import income and mode with giftermidens/deficive/genous; import combot-filed lates; import income for control and lates and income for c			*	·	*
correct control to appear access and employment and the description of the control to the contro			* @naram entity the Formulario Aditivo Agencia entity to		* @author Sistema SARSR
** descrives to the Ablegio Constructors: Import formore Ablegio Constructors: Import			1		_
introut introduct. Self-likes: unance international control co				, ,	=
import tomorbo. Getter; import tomorbo. More (Constructors) import		·	_		_
Import combook. basing Controvalurally (import combook. basing Controvalural) (import combook. b				l · · · · · · · · · · · · · · · · · · ·	^/
of communication control contr					
Security of the properties of the security of			l '	*/	
Gibbling-Construction Gibling-Construction Gibbling-Construction G		import lombok.Setter;	toFormularioAditivoDTO(FormularioAditivoAgencia entity)		
General Content		@Builder	{		/**
Gisterier Gister		@AllArgsConstructor	return FormularioAditivoDTO.builder()		* Converte uma entidade FormularioAditivoAgencia para
Genter public class Formularial dislocation () { private language public class Formularial dislocat		@NoArgsConstructor	.id(entity.getId())		FormularioAditivoDTO.
Getter public class primulatin-distroyOTT { private lang sit, private lang sit, private lang sit, private part of the state of the stat		@Getter	.numeroRecibo(entity.getNumeroRecibo())	/**	*
position clares formula anochatino (CDI) private Largi (at many filter Largi (at private L			, , , , , , , , , , , , , , , , , , , ,	* Converts a FormularioAditivoAgencia entity	* @param entity a entidade FormularioAditivoAgencia para
privete Long (1) privete Long (1) privete Long (1) privete String troph/gen/claffequilidations; privete String troph/gen/claffequilidations; privete String troph/gen/claffequilidations) privete String		-	cnniAgenciaReguladora(entity getAgenciaReguladora() g	-	, · · · · · · · · · · · · · · · · · · ·
private Long numeroteolocy private Date data fravity private Date data		1 '		*	
private String can federal activation of the production of the pro			1	* @naram entity the	=
private Dark dataFrivor, public FormularinAdditivoDTO toFormularinAdditivoDTO (incomplainAdditivoDTO) toFormularinAdditivoDTO (incomplainAdditivoDTO)				1 · · · · · · · · · · · · · · · · · · ·	
but for ormular in-Additive DTO (Formular in			.buita();	-	
InformularioAdditiveOTO (primularioAdditiveOTO (builder)			}		
return formularioAddrivoDTO builder() distingty,gadd(s) distingty,gad(s) disting					·
dicentity,getIdu(f)duterroreRecibo(cntity,getMumerRecibo(f))dutd(f):duterroreRecibo(cntity,getMumerRecibo(f))dutd(f):duterroreRecibo(cntity,getMumerRecibo(f))dutd(f):duterroreRecibo(cntity,getMumerRecibo(f))dutd(f):duterroreRecibo(cntity,getMumerRecibo(f))duterroreRecibo(cntity,getMumerRecibo(f))duterroreRecibo(cntity,getMumerRecibo(f))duterroreRecibo(cntity,getMumerRecibo(f))duterroreRecibo(finity,getMumerRecibo(finity,getMu		, , , , , , , , , , , , , , , , , , , ,		1 -	l '
enclis entity (npi/agencia/Reguladoral Reguladoral (, getCnpi())dataEnvio(entity, getDataEnvio())build():build		return FormularioAditivoDTO.builder()		, ·	
		.id(entity.getId())		toFormularioAditivoDTO(FormularioAditivoAg	if (entity == null) {
detatEmio(antity.getDataEmio(l))build(l); } // Fim do arquivo: FormularioAditivoDTO.java // Arquivo: FormularioAditivoDo.java // Arquivo: Java // Arquivo: FormularioAditivoDo.java // Arquivo: FormularioAditivoDo.java // Arquivo: FormularioAditivoDo.java // Arquivo: Java // Arquivo: FormularioAditivoDo.java // Arquivo: FormularioAditivoDo.java // Arquivo: Java // Arquivo: FormularioAditivoDo.java // Arquivo: FormularioAditivoDo.java // Arquivo: FormularioAditivoDo.java // Arquivo: Java // Arquivo: FormularioAditivoDo.java // Arquivo: FormularioAditivoDo.java // Arquivo: Java // Arquivo: FormularioAditivoDo.java // Arquivo: Java // Arquivo: FormularioAditivoDo.java // Arquivo: Java // Arquiv		.numeroRecibo(entity.getNumeroRecibo())		encia entity) {	throw new
detatEmio(antity.getDataEmio(l))build(l); } // Fim do arquivo: FormularioAditivoDTO.java // Arquivo: FormularioAditivoDo.java // Arquivo: Java // Arquivo: FormularioAditivoDo.java // Arquivo: FormularioAditivoDo.java // Arquivo: FormularioAditivoDo.java // Arquivo: Java // Arquivo: FormularioAditivoDo.java // Arquivo: FormularioAditivoDo.java // Arquivo: Java // Arquivo: FormularioAditivoDo.java // Arquivo: FormularioAditivoDo.java // Arquivo: FormularioAditivoDo.java // Arquivo: Java // Arquivo: FormularioAditivoDo.java // Arquivo: FormularioAditivoDo.java // Arquivo: Java // Arquivo: FormularioAditivoDo.java // Arquivo: Java // Arquivo: FormularioAditivoDo.java // Arquivo: Java // Arquiv		.cnpjAgenciaReguladora(entity.getAgenciaReguladora().getCnpj())		return FormularioAditivoDTO.builder()	IllegalArgumentException("FormularioAditivoAgencia não pode
				.id(entity.getId())	
.numeroRecibo(entity.getNumeroRecibo(n) // Firm do arquive; FormularioAdritvoDTO.java // Arquive; FormularioSmaruDTO.java // Caminho: thr.gov/ana/sarashumodel.dto/agenciareguladora/adritvo\FormularioSmaruDTO.java package br.gov.ana.sarshumodel.dto/agenciareguladora/adritvo\FormularioSmaruDTO.java package br.gov.ana.sarshumodel.dto/agenciareguladora/adritvo\FormularioSmaruDTO.java package br.gov.ana.sarshumodel.dto/agenciareguladora/adritvo\FormularioAdritvoSmaru; import br.gov.ana.sarshumodel.entity, FormularioAdritvoSmaru; import private Long (d. gov.ana.sarshumodel.entity, FormularioAdritvoSmaru; import private Long (d. gov.ana.sarshumodel.entity, FormularioAdritvoSmaru) gov.ana.sarshumodel.entity, getVagenciaReguladora(), JetCnp()() .dataEmvio(entity.getDataEmvio()) .build(); } return FormularioAdritvoDTO.builder() .identity.getAgenciaReguladora(), getCnp()() .dataEmvio(entity.getDataEmvio()) .build(); } roturn FormularioAdritvoDTO.getDataEmvio() .build(); } roturn FormularioAdritvoTor identity, getLagenciaReguladora(), getCnp()() .dataEmvio(entity.getLagenciaReguladora(), getCnp()() .dataEmvio(entity.getLagenciaReguladora(), getCnp()() .dataEmvio(entity.getLagenciaReguladora(), getCnp()() .dataEmvio(entity.getLagenciaReguladora(), getCnp()() .dataEmvio(entity.getLagenciaReguladora(), getCnp()() .dataEmvio(entity.getLagenciaReguladora() .dataEmvio(entity.getLagen				7 6 7 6 7 6 7 6 7 6 7 6 7 6 7 6 7 6 7 6	}
// Firm do arquivo: FormularioAditivoDTO.java // FormularioAditivoDTO.java // Aquivo: FormularioAditivoDTO.java // Caminito the food analysis of the food an		1 }		numeroRecibo(entity.getNumeroRecibo())	, ,
// Firm do arquivo: FormularioAndtivoDTO.java // Arquivo: FormularioSmstrDTO.java // Arquivo: FormularioSmstrDTO.java // Caminho:		11		amoronooibo(onticy.goti variioronooibo())	return Formulario∆ditivoDTO builder()
// Arquivo: FormularioSmaruDTO,java guladora(),getCnp(i) .dataEnvio(entity,getDataEnvio()) .build(); .build(); .dataEnvio(entity,getDataEnvio()) .build(); .b		// Eim de arquive: Formularie Aditive DTO iove		anni Aganaia Paguladara/antitu gat Aganaia Pa	
// Carninto:		1			
horgovianoisarsthimodelytolagenciareguladoraaditivo/FormularioSmsruDTO.java package by gowana.asarbamodel duta agencianeguladora aditivo; import by gowana.asarbamodel duta agencianeguladora aditivo; import by gowana.asarbamodel entity.FormularioAditivoAgencia; import by gowana.asarbamodel.entity.FormularioAditivoAgencia; import by gowana.asarbamodel.entity.FormularioAditivoAgencia; import lombolc*; import lombo					.numerokecibo(entity.getNumerokecibo())
package br.gov.ana.sarsb.model.atto.agenciareguladora.aditivo; import br.gov.ana.sarsb.model.entity.FormularioAditivoSmsru; import br.gov.ana.sarsb.model.entity.FormularioAditivoSmsru @Bullder @Bulldrage @Bulldrage @Bulldrage @Bulldrage @Setter @Setter gyravete String numeroRecibo; private Long deprefeturia; private Dadi gotalstrivio; public FormularioSmsruDTO toFormularioSmsruDTO unimeroRecibo(String valueOf(entity.gettPrefeturia, gettPrefeturia, gettPrefeturia, deltentity.gettPrefeturia, deltentity.gettPrefeturi				, , , , , , , , , , , , , , , , , , , ,	
import brgwana sarab.model.entity.FormularioAditivoSmsru; import lombok *; import lombok *; import lambok *;				.build();	
import brigovana.sarsb.model.entity.FormularioAditivoSmsru; import import brimbok.*; import import brimbok.*; import plava.util.Date; @Bulider @AllArgsConstructor @NoArgsConstructor @Getter @Satter public class FormularioSmsruDTO { private Long id; private Long id; private Long id; private Long id; private Date dataEnvio; private Date dataEnvio; private Date dataEnvio; public FormularioAditivoSmsru entity) { return FormularioSmsruDTO.bulider()				}	
import lombok.*; import jaw util. Date; @Builder @AllArgs Constructor @AllArgs Constructor @Getter @Setter public class FormularioSmsruDTO { private Long id; private Long id; private Long id; private Date dataEnvio; public FormularioSmsruDTO toFormularioAditivoDTO(FormularioAditivoSmsru entity) { return FormularioAditivoSmsruDTO.builder()		import br.gov.ana.sarsb.model.entity.FormularioAditivoAgencia;			.build();
import java util. Date; @Bullder @AllArgs Constructor @NoArgs Constructor @Getter @Setter grivate Long (id;		import br.gov.ana.sarsb.model.entity.FormularioAditivoSmsru;			}
@BulldargsConstructor @AllargsConstructor @NoArgsConstructor @Getter @Setter public class FormularioSmsruDTO { private Longid; private Longid; private Longid; private Longidprefeitura; private Longidprefeitura; private Date dataEnvio; public FormularioAsmsruDTO toFormularioAditivoSmsru entity) { return FormularioSmsruDTO.builder()		import lombok.*;			
@AllArgsConstructor @NoArgsConstructor @Getter @Setter public class FormularioSmsruDTO { private Long id; private Long id; private Date dataEnvio; private Date dataEnvio; public FormularioSmsruDTO toFormularioAdditivoDTO(FormularioAdditivoSmsru entity) { return FormularioSmsruDTO.builder()		import java.util.Date;			
@AllArgsConstructor @NoArgsConstructor @Getter @Setter public class FormularioSmsruDTO { private Long id; private Long id; private Date dataEnvio; private Date dataEnvio; public FormularioSmsruDTO toFormularioAdditivoDTO(FormularioAdditivoSmsru entity) { return FormularioSmsruDTO.builder()		@Builder			
@NoArgsConstructor @Getter @Setter public class FormularioSmsruDTO { private Long id; private String numeroRecibo; private Long idprefeitura; private Long idprefeitura; private Long idprefeitura; private Date dataEnvio; public FormularioSmsruDTO toFormularioAditivoDT(FormularioAditivoSmsru entity) { return FormularioSmsruDTO.builder()					
@Getter @Setter public class FormularioSmsruDTO { private Long id; private Long id; private Long idprefeitura; private Long idprefeitura; private Date dataEnvio; public FormularioSmsruDTO toFormularioAditivoSmsru entity) { return FormularioSmsruDTO.builder()					
@Setter public class FormularioSmsruDTO { private Long id; private String numeroRecibo; private Long idprefeitura; private Long idprefeitura; private Long idprefeitura; private Date dataEnvio; public FormularioSmsruDTO toFormularioAditivoDTO(FormularioAditivoSmsru entity) { return FormularioSmsruDTO.builder()					
public class FormularioSmsruDTO { private Long id; private String numeroRecibo; private Long idprefeitura; private Date dataEnvio; public FormularioSmsruDTO toFormularioAditivoDTO(FormularioAditivoSmsru entity) { return FormularioAssruDTO.builder()					
private Long id; private Long idprefeitura; private Long idprefeitura; private Date dataEnvio; public FormularioSmsruDTO toFormularioAditivoDTO(FormularioAditivoSmsru entity) { return FormularioSmsruDTO.builder()		=			
private String numeroRecibo; private Long idprefeitura; private Date dataEnvio; public FormularioSmsruDTO toFormularioAditivoDTO(FormularioSmsruDTO.builder()		,			
private Long idprefeitura; private Date dataEnvio; public FormularioSmsruDTO toFormularioAditivoSTO(FormularioAditivoSmsru entity) { return FormularioSmsruDTO.builder()					
private Date dataEnvio; public FormularioSmsruDTO toFormularioAditivoDTO(FormularioAditivoSmsru entity) { return FormularioSmsruDTO.builder()					
public FormularioSmsruDTO toFormularioAditivoSmsru entity) { return FormularioSmsruDTO.builder()					
toFormularioAditivoDTO(FormularioAsmsru entity) { return FormularioSmsruDTO.builder() .id(entity.getId()) .numeroRecibo(String.valueOf(entity.getNumeroRecibo())) .idprefeitura(entity.getPrefeitura().getId()) // Use o ID da prefeitura corretamente .dataEnvio(new Date()) // Define a data atual como data de envio .build(); }					
return FormularioSmsruDTO.Duilder() .id(entity.getId()) .numeroRecibo(String.valueOf(entity.getNumeroRecibo())) .idprefeitura(entity.getPrefeitura().getId()) // Use o ID da prefeitura corretamente .dataEnvio(new Date()) // Define a data atual como data de envio .build(); }		· ·			
.id(entity.getId()) .numeroRecibo(String.valueOf(entity.getNumeroRecibo())) .idprefeitura(entity.getPrefeitura().getId()) // Use o ID da prefeitura corretamente .dataEnvio(new Date()) // Define a data atual como data de envio .build(); }					
.id(entity.getId()) .numeroRecibo(String.valueOf(entity.getNumeroRecibo())) .idprefeitura(entity.getPrefeitura().getId()) // Use o ID da prefeitura corretamente .dataEnvio(new Date()) // Define a data atual como data de envio .build(); }		return FormularioSmsruDTO.builder()			
.numeroRecibo(String.valueOf(entity.getNumeroRecibo())) .idprefeitura(entity.getPrefeitura().getId()) // Use o ID da prefeitura corretamente .dataEnvio(new Date()) // Define a data atual como data de envio .build(); }					
.idprefeitura(entity.getPrefeitura().getId()) // Use o ID da prefeitura corretamente .dataEnvio(new Date()) // Define a data atual como data de envio .build(); }		, , , , , , , , , , , , , , , , , , , ,			
prefeitura corretamente .dataEnvio(new Date()) // Define a data atual como data de envio .build(); }					
data atual como data de envio build(); }					
data atual como data de envio .build(); } }					
.build(); } }					
}					
} // Fim do arquivo: FormularioSmsruDTO.java		.build();			
} // Fim do arquivo: FormularioSmsruDTO.java		· · · · · · · · · · · · · · · · · · ·			
// Fim do arquivo: FormularioSmsruDIO.java		1			
		// Fim do arquivo: FormularioSmsruDTO.java			

ID	Original Documentation A	Documentation - LLM B	Documentation - LLM C	Documentation - LLM D
	Original code without LLM modification	Code with LLM change	Code with LLM change	Code with LLM change
9	// Arquivo: InstrumentoCobrancaTemEtapasDTO.java	/**	/**	/**
	// Caminho:	* Loads the detailed description of the service stage.	* [Melhoria Amazon Q + ChatGPT]	* DTO para etapas de instrumentos de cobrança.
	\br\gov\ana\sarsb\model\dto\municipio\lnstrumentoCobrancaTemEtapasDTO.java	*	* Este pacote foi revisado e expandido	* Fornece detalhamento das etapas de serviço.
		* @return the description of the service stage type	automaticamente para garantir maior	*
	package br.gov.ana.sarsb.model.dto.municipio;	*/	completude (ΔC), clareza (ΔH) e precisão	* @author Sistema SARSB
		public String carregaDetalhamento() {	factual (ΔT).	* @version 1.0
	import br.gov.ana.sarsb.model.entity.TipoEtapaServico;	return tipoEtapaServico.getDescricao();	* As descrições foram enriquecidas com	* @since 1.0
	import lombok.AllArgsConstructor;	}	contexto de uso, dependências e finalidade	*/
	import lombok.Builder; import lombok.Getter;		arquitetural. */	
	import torribok.Getter; import tombok.NoArgsConstructor;		"7	
	import tombok.NoArgsconstructor,			/**
	import tombok.Setter,			* Carrega a descrição detalhada da etapa de serviço.
				*
	@Builder		/**	* @return a descrição do tipo de etapa de serviço, ou string vazia
	@AllArgsConstructor		* Loads the detailed description of the	se tipoEtapaServico for {@code null}
	@NoArgsConstructor		service stage.	* @since 1.0
	@Getter		*	*/
	@Setter		* @return the description of the service	public String carregaDetalhamento() {
	public class InstrumentoCobrancaTemEtapasDTO {		stage type	return tipoEtapaServico != null ?
			*/	tipoEtapaServico.getDescricao(): "";
	private TipoEtapaServico tipoEtapaServico;		public String carregaDetalhamento() {	}
l			return tipoEtapaServico.getDescricao();	
	public String carregaDetalhamento() {		}	
	return tipoEtapaServico.getDescricao();			
	}			
l				
l				
l	// Fim do arquivo: InstrumentoCobrancaTemEtapasDTO.java			

ID Original Documentation A	Documentation - LLM B	Documentation - LLM C	Documentation - LLM D
Original code without LLM modification	Code with LLM change	Code with LLM change	Code with LLM change
10 // Arquivo: SecurityResource.java	/**	/**	/**
// Caminho: \br\gov\ana\sars\config\security\boundary\SecurityResource.java package br.gov.ana.sars\bconfig.security.boundary; import java.util.Collection; import org.springframework.security.access.prepost.PreAuthorize;	* Retrieves user information from the authenticated user's session. * *@param authentication the authentication object	* [Melhoria Amazon Q + ChatGPT] * Este pacote foi revisado e expandido automaticamente para garantir maior completude (ΔC), clareza (ΔH) e precisão	* Recurso REST para informações do usuário autenticado. * Fornece endpoints para consulta de dados do usuário logado.
import org.springframework.security.core.Authentication; import org.springframework.security.core.GrantedAuthority; import org.springframework.web.bind.annotation.GetMapping; import org.springframework.web.bind.annotation.ResponseBody; import org.springframework.web.bind.annotation.RestController;	containing user details * @return UserInfoDTO with user's name, email, and username, or null if user is not a CustomUserDTO */ @GetMapping(produces = MediaType.APPLICATION_JSON_VALUE)	factual (ΔT). * As descrições foram enriquecidas com contexto de uso, dependências e finalidade arquitetural. */	* * @author Sistema SARSB * @version 1.0 * @since 1.0 */
<pre>@RestController @RequestMapping(path = "/security") @PreAuthorize("isAuthenticated()") public class SecurityResource { /**</pre>	<pre>public UserInfoDTO getUserInfo(Authentication authentication) { var userDetails = (UserDetails) authentication.getPrincipal(); if(userDetails instanceof CustomUserDTO) {</pre>	/** * Retrieves user information from the	/** * Recupera informações do usuário a
* Retrieves the collection of granted authorities (profiles/roles) for the currently authenticated user.	,	authenticated user's session.	partir da sessão do usuário autenticado.
* This endpoint returns the security authorities associated with the authenticated user's sess * *@param authentication the authentication object containing user credentials and authoritie *@return a collection of granted authorities representing the user's profiles/roles */	sion. return UserInfoDTO.builder() .name(customUser.getFullName())	* @param authentication the authentication object containing user details * @return UserInfoDTO with user's name, email, and username, or null if user is not a	* @param authentication o objeto de autenticação contendo detalhes do usuário * @return UserInfoDTO com nome, email e username do usuário, ou {@code
@GetMapping(value = "/currentuser/profiles") @ResponseBody public Collection extends GrantedAuthority profiles (Authentication authentication) {	.build(); }	CustomUserDTO */ @GetMapping(produces =	null} se o usuário não for CustomUserDTO * @since 1.0
return authentication.getAuthorities(); } // Fim do arquivo: SecurityResource.java	return null; }	MediaType.APPLICATION_ISON_VALUE) public UserInfoDTO getUserInfo(Authentication authentication) {	*/ @GetMapping(produces = MediaType.APPLICATION_JSON_VALUE) public UserInfoDTO
// Arquivo: UserResource.java // Caminho: \br\gov\ana\sarsb\config\security\boundary\UserResource.java package br.gov.ana.sarsb.config.security.boundary; import org.springframework.http.MediaType; import org.springframework.security.access.prepost.PreAuthorize; import org.springframework.security.core.Authentication; import org.springframework.security.core.userdetails.UserDetails; import org.springframework.web.bind.annotation.GetMapping; import org.springframework.web.bind.annotation.RequestMapping; import org.springframework.web.bind.annotation.RestController;		var userDetails = (UserDetails) authentication.getPrincipal(); if(userDetails instanceof CustomUserDTO) { CustomUserDTO customUser = (CustomUserDTO) userDetails; return UserInfoDTO.builder()	getUserInfo(Authentication authentication) { var userDetails = (UserDetails) authentication.getPrincipal(); if(userDetails instanceof CustomUserDTO) { CustomUserDTO customUser = (CustomUserDTO) userDetails; return UserInfoDTO.builder()
import br.gov.ana.sarsb.config.security.dto.CustomUserDTO; import br.gov.ana.sarsb.config.security.dto.UserInfoDTO; @RestController @RequestMapping(path = "/userinfo") @PreAuthorize("isAuthenticated()")		.preferredUsername(customUser.getUsern ame()) .build(); }	.name(customUser.getFullName()) .email(customUser.getMail()) .preferredUsername(customUser.getUse rname())
public class UserResource { @GetMapping(produces = MediaType.APPLICATION_JSON_VALUE) public UserInfoDTO getUserInfo(Authentication authentication) { var userDetails = (UserDetails) authentication.getPrincipal(); if(userDetails instanceof CustomUserDTO) { CustomUserDTO customUser = (CustomUserDTO) userDetails; return UserInfoDTO.builder()	ne())	return null; }	.build(); } return null; }
return null; }			
} // Fim do arquivo: UserResource.java			

```
Original Documentation A
Original code without LLM modification
// Arquivo: AjaxLogoutSuccessHandler.java
// Caminho: \br\gov\ana\sarsb\config\security\handler\AjaxLogoutSuccessHandler.java
package br.gov.ana.sarsb.config.security.handler:
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServletRequest:
import javax.servlet.http.HttpServletResponse;
import org.apache.commons.lang3.StringUtils;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.security.core.Authentication;
import org.springframework.security.oauth2.provider.token.TokenStore;
import org.springframework.security.web.authentication.AbstractAuthenticationTargetUrlRequestHandler:
import org.springframework.security.web.authentication.logout.LogoutSuccessHandler;
import org.springframework.stereotype.Component;
@Component
public class AjaxLogoutSuccessHandler extends AbstractAuthenticationTargetUrlRequestHandler
   implements LogoutSuccessHandler {
 public static final String BEARER_AUTHENTICATION = "Bearer";
 private static final String HEADER AUTHORIZATION = "authorization":
 @Autowired
 private TokenStore tokenStore;
  * Handles successful logout by invalidating OAuth2 tokens and setting appropriate HTTP response status.
  * This method extracts the Bearer token from the Authorization header, removes both the access token
  * and its associated refresh token from the token store, and returns an HTTP 200 OK status.
  * @param request the HTTP servlet request containing the authorization header with Bearer token
  * @param response the HTTP servlet response that will be set to status 200 OK
  * @param authentication the authentication object representing the user being logged out
  * @throws IOException if an input or output exception occurs during request/response processing
  * @throws ServletException if a servlet-specific exception occurs during processing
 @Override
 public void onLogoutSuccess(HttpServletRequest request, HttpServletResponse response, Authentication authentication)
     throws IOException, ServletException {
   String token = request.getHeader(HEADER_AUTHORIZATION):
   if (token != null && token.startsWith(BEARER_AUTHENTICATION)) {
     final var oAuth2AccessToken = tokenStore
         .readAccessToken(StringUtils.substringAfter(token, BEARER_AUTHENTICATION));
     if (oAuth2AccessToken != null) {
       tokenStore.removeAccessToken(oAuth2AccessToken);
       if (oAuth2AccessToken.getRefreshToken() != null) {
           tokenStore.removeRefreshToken(oAuth2AccessToken.getRefreshToken());
   response.setStatus(HttpServletResponse.SC OK):
// Fim do arquivo: AjaxLogoutSuccessHandler.java
```

Documentation - LLM B

Code with LLM change

```
* Handles successful logout by invalidating OAuth2 tokens and setting appropriate HTTP response status.

*
```

 * This method extracts the Bearer token from the Authorization header, removes both the access token

 * and its associated refresh token from the token store, and returns an HTTP 200 OK status.

 ${}^{\star}\,@param\,request\,the\,HTTP\,servlet\,request\,containing\,the\,authorization\,header\,with\,Bearer\,token$

* @param response the HTTP servlet response that will be set to status 200 OK

* @param authentication the authentication object representing the user being logged out

* @throws IOException if an input or output exception occurs during request/response processing

 $\hbox{* @throws Servlet-Exception if a servlet-specific exception occurs during processing}$

@Override

```
public void onLogoutSuccess(HttpServletRequest request, HttpServletResponse response, Authentication authentication)
```

```
Authentication authentication)
throws IOException, ServletException {
String token = request.getHeader(HEADER_AUTHORIZATION);
if (token != null && token.startsWith(BEARER_AUTHENTICATION)) {
final var oAuth2AccessToken = tokenStore
.readAccessToken(StringUtils.substringAfter(token, BEARER_AUTHENTICATION));
if (oAuth2AccessToken != null) {
    tokenStore.removeAccessToken(oAuth2AccessToken);
    if (oAuth2AccessToken.getRefreshToken() != null) {
        tokenStore.removeRefreshToken(oAuth2AccessToken.getRefreshToken());
    }
}
response.setStatus(HttpServletResponse.SC_OK);
```

Original Documentation C Documentation – LLM D Original code without LLM modification Code with LLM change * [Melhoria Amazon Q + ChatGPT] * Manipulador de sucesso de logout para requisições AJAX. * Este pacote foi revisado e expandido automaticamente para garantir maior completude (ΔC), clareza (ΔH) e precisão factual (ΔT). * Gerencia invalidação de tokens OAuth2 durante o processo de logout. * As descrições foram enriquecidas com contexto de uso, dependências e finalidade arquitetural. * @author Sistema SARSB * @version 1.0 * @since 1.0 */ * Handles successful logout by invalidating OAuth2 tokens and setting appropriate HTTP response status. * This method extracts the Bearer token from the Authorization header, removes both the access token * and its associated refresh token from the token store, and returns an HTTP 200 OK status. * Manipula logout bem-sucedido invalidando tokens OAuth2 e definindo status HTTP apropriado. * @param request the HTTP servlet request containing the authorization header with Bearer token * Este método extrai o token Bearer do cabeçalho Authorization, remove tanto o token de acesso * @param response the HTTP servlet response that will be set to status 200 OK * quanto seu token de refresh associado do armazenamento de tokens, e retorna status HTTP 200 * @param authentication the authentication object representing the user being logged out OK. * @throws IOException if an input or output exception occurs during request/response processing * @param request a requisição HTTP servlet contendo o cabeçalho de autorização com token * @throws ServletException if a servlet-specific exception occurs during processing @Override * @param response a resposta HTTP servlet que será definida com status 200 OK * @param authentication o objeto de autenticação representando o usuário sendo deslogado public void onLogoutSuccess(HttpServletRequest request, HttpServletResponse response, Authentication authentication) throws IOException, ServletException { * @throws IOException se ocorrer exceção de entrada ou saída durante processamento de String token = request.getHeader(HEADER_AUTHORIZATION); request/response if (token != null && token.startsWith(BEARER_AUTHENTICATION)) { * @throws ServletException se ocorrer exceção específica de servlet durante processamento final var oAuth2AccessToken = tokenStore .readAccessToken(StringUtils.substringAfter(token, BEARER_AUTHENTICATION)); if (oAuth2AccessToken != null) { @Override tokenStore.removeAccessToken(oAuth2AccessToken); public void onLogoutSuccess(HttpServletRequest request, HttpServletResponse response, if (oAuth2AccessToken.getRefreshToken() != null) { Authentication authentication) tokenStore.removeRefreshToken(oAuth2AccessToken.getRefreshToken()); throws IOException, ServletException { String token = request.getHeader(HEADER_AUTHORIZATION); if (token != null && token.startsWith(BEARER AUTHENTICATION)) { final var oAuth2AccessToken = tokenStore response.setStatus(HttpServletResponse.SC OK); .readAccessToken(StringUtils.substringAfter(token, BEARER AUTHENTICATION)); if (oAuth2AccessToken != null) { tokenStore.removeAccessToken(oAuth2AccessToken); if (oAuth2AccessToken.getRefreshToken() != null) { tokenStore.removeRefreshToken(oAuth2AccessToken.getRefreshToken()): response.setStatus(HttpServletResponse.SC OK);