

Estimation Story Points – Projet Spring Boot

Estimation Story Points

Projet Spring Boot



Jira

Agile



Scrum



Poker
fibonnaci

Dr. BADR EL KHALYLY

1. CRUD – Create, Read, Update, Delete

Opération	Contexte	Story Points	Détails
Create	Création simple avec DTO	2	@PostMapping, mapping DTO vers entité, service
	Avec validations (@Valid)	3–5	Unicité, règles métier
	Création avec fichiers	5–8	MultipartFile, stockage
	Création avec cascade (@OneToMany)	5–8	Enregistrement d'objets liés
Read	Lecture par ID	1–2	findById, gestion 404
	Lecture de liste simple	2	findAll(), mapping DTO
	Recherche avec filtres	5–8	Spring Specifications / Criteria
	Lecture avec entités liées	3–5	@EntityGraph, mapping enrichi
Update	Mise à jour simple (PUT)	3	Vérification, mise à jour, sauvegarde
	Mise à jour partielle (PATCH)	5	Champs optionnels, logique conditionnelle
	Mise à jour avec validations métier	5–8	Règles métier complexes
	Mise à jour avec audit (updatedBy, etc.)	5	Listener ou AOP
Delete	Suppression simple	1	deleteById, gestion exception
	Suppression logique (soft delete)	3–5	Champ deleted, filtre global
	Suppression avec vérification de dépendances	5–8	Vérification via Repository
	Suppression en cascade	5–8	Entités liées, suppression contrôlée

2. Pagination, Tri, Filtrage

Story	Story Points	Détails
Pagination simple (Pageable)	2	PageRequest, Page<T>
Pagination + tri (sort)	3	Ajout de paramètres dynamiques
Pagination + filtres complexes	5–8	Construction dynamique (Specifications)

3. Authentification et Sécurité

Story	Story Points	Détails
Authentification avec JWT	5	Génération token, endpoint login
Rôles / @PreAuthorize	5–8	Sécurité sur les endpoints
Refresh Token	5	Renouvellement JWT
Intégration OAuth2 / LDAP	8–13	Configuration avancée
Extraction des rôles depuis le token pour logique métier	5	JwtDecoder, injection SecurityContext
Sécurisation de l'API Gateway avec token	5–8	Interception globale, header check

4. GitLab CI/CD

Story	Story Points	Détails
Mise en place de .gitlab-ci.yml	3	Build, test, stages
Déploiement automatique vers dev/staging	5	SSH keys, Docker Registry
Intégration de SonarQube ou Quality Gate	5–8	Analyse qualité de code
Pipeline avancé avec Trivy / OWASP check	8–13	SAST / container scan
Déploiement sur Kubernetes (EKS, GKE, etc.)	13+	Helm, secrets, autoscaling
Notifications Slack/Teams dans la CI	3	Webhooks, tokens API

5. Caching (Spring Cache)

Story	Story Points	Détails
@Cacheable simple (cache mémoire local)	2	Clé simple, lecture uniquement
@CacheEvict après update/delete	3	Invalidation cache
Cache TTL, eviction policy (Ehcache, Caffeine)	5	Configuration avancée
Caching conditionnel	3–5	unless, condition
Cache distribué avec Redis	5–8	Spring Boot Redis
Logger les hits/miss de cache	5–8	CacheManager, AOP
Tests unitaires du cache	3	Mockito, vérification de hit

6. Transactions

Story	Story Points	Détails
@Transactional simple	2	Rollback automatique si exception
Transactions imbriquées (REQUIRES_NEW)	5	Propagation personnalisée
Gestion manuelle avec TransactionTemplate	5	Contrôle programmatique
Rollback conditionnel	5–8	noRollbackFor, exceptions spécifiques
Test d'intégration de rollback	3–5	DB H2, simulation d'erreurs
Audit ou logging de transaction	5–8	AOP, @Before / @After transactionnels

7. Feign Client

Story	Story Points	Détails
Déclaration d'un @FeignClient simple (JSON)	2	GET/POST avec DTO simple
Feign avec configuration (timeout, log)	3–5	Custom config ou application.yml

Story	Story Points	Détails
Feign avec JWT Authorization	5	RequestInterceptor personnalisé
@FeignClient avec fallback (fallbackFactory)	5–8	Implémentation de classe fallback
Feign avec XML (decoder personnalisé)	5–8	XmlMapper, Decoder custom
Feign pour multipart/form-data	5	Upload de fichiers
Feign avec Basic Auth ou OAuth2	5–8	Header personnalisés ou token manager
Tests MockWebServer ou WireMock	3–5	Simuler réponses HTTP
Logger des appels Feign	3	Interceptor + logs HTTP
Feign + Resilience4j retry/circuit breaker	5	Retry 3x, délai exponentiel
Partage de DTO via librairie commune	2	Gradle/Maven module partagé

8. Gestion centralisée des exceptions avec @ControllerAdvice

User Story Technique	Story Points	Détails
Mettre en place un @ControllerAdvice global	2	Classe annotée avec @ControllerAdvice, gestion d'exception simple (ex : IllegalArgumentException)
Gérer les exceptions personnalisées (CustomException)	3	Création de classes d'exception (ex : BusinessException, UnauthorizedException)
Retourner des objets structurés (ErrorResponse, timestamp, path)	3–5	Structure d'erreur uniforme avec code, message, timestamp, etc.
Mapper différents codes HTTP (400, 404, 409, 500, 401)	5	Utilisation de @ExceptionHandler + @ResponseStatus
Extraction du message depuis une ConstraintViolationException	3–5	Traitement des violations de validation de Bean (javax)
Gestion des erreurs JWT (ExpiredJwtException, SignatureException)	5	Intercepter les erreurs liées au token dans le contrôleur global

User Story Technique	Story Points	Détails
Localisation (i18n) des messages d'erreur	5–8	Fichiers messages.properties, MessageSource, injection dans la réponse
Logging conditionnel des erreurs techniques / métier	5	Utilisation de Logger, stratégie de log différenciée
Test des cas d'erreur avec MockMvc	3	Tests unitaires pour les cas d'exception avec vérification du corps de réponse JSON
Masquage des erreurs internes (ex : stacktrace) pour l'utilisateur	3–5	Masquage des messages techniques, affichage de messages lisibles

Notes complémentaires

- Les estimations suivent l'échelle de Fibonacci : **1, 2, 3, 5, 8, 13, ...**
- Ajoute +1 à +2 SP si :
 - L'opération implique **sécurité / tokens JWT**
 - L'appel est **à travers une API Gateway**
 - Il y a **du mapping complexe DTO ↔ Entity**