



Numéro d'étudiant :

<input type="checkbox"/> 0	<input type="checkbox"/> 0	<input type="checkbox"/> 0	<input type="checkbox"/> 0	<input type="checkbox"/> 0	<input type="checkbox"/> 0	<input type="checkbox"/> 0	<input type="checkbox"/> 0
<input type="checkbox"/> 1	<input type="checkbox"/> 1	<input type="checkbox"/> 1	<input type="checkbox"/> 1	<input type="checkbox"/> 1	<input type="checkbox"/> 1	<input type="checkbox"/> 1	<input type="checkbox"/> 1
<input type="checkbox"/> 2	<input type="checkbox"/> 2	<input type="checkbox"/> 2	<input type="checkbox"/> 2	<input type="checkbox"/> 2	<input type="checkbox"/> 2	<input type="checkbox"/> 2	<input type="checkbox"/> 2
<input type="checkbox"/> 3	<input type="checkbox"/> 3	<input type="checkbox"/> 3	<input type="checkbox"/> 3	<input type="checkbox"/> 3	<input type="checkbox"/> 3	<input type="checkbox"/> 3	<input type="checkbox"/> 3
<input type="checkbox"/> 4	<input type="checkbox"/> 4	<input type="checkbox"/> 4	<input type="checkbox"/> 4	<input type="checkbox"/> 4	<input type="checkbox"/> 4	<input type="checkbox"/> 4	<input type="checkbox"/> 4
<input type="checkbox"/> 5	<input type="checkbox"/> 5	<input type="checkbox"/> 5	<input type="checkbox"/> 5	<input type="checkbox"/> 5	<input checked="" type="checkbox"/> 5	<input type="checkbox"/> 5	<input type="checkbox"/> 5
<input type="checkbox"/> 6	<input type="checkbox"/> 6	<input type="checkbox"/> 6	<input type="checkbox"/> 6	<input type="checkbox"/> 6	<input type="checkbox"/> 6	<input type="checkbox"/> 6	<input type="checkbox"/> 6
<input type="checkbox"/> 7	<input type="checkbox"/> 7	<input type="checkbox"/> 7	<input type="checkbox"/> 7	<input type="checkbox"/> 7	<input type="checkbox"/> 7	<input type="checkbox"/> 7	<input type="checkbox"/> 7
<input type="checkbox"/> 8	<input type="checkbox"/> 8	<input type="checkbox"/> 8	<input type="checkbox"/> 8	<input type="checkbox"/> 8	<input type="checkbox"/> 8	<input type="checkbox"/> 8	<input checked="" type="checkbox"/> 8
<input type="checkbox"/> 9	<input type="checkbox"/> 9	<input type="checkbox"/> 9	<input type="checkbox"/> 9	<input type="checkbox"/> 9	<input type="checkbox"/> 9	<input checked="" type="checkbox"/> 9	<input type="checkbox"/> 9

Département: SI

Durée: 60min (+20min tiers-temps)

Documents: 1 feuille manuscrite

Nom: MARVINPrénom: ANNE

Noircissez ■ ou cochez ☒ les réponses de manière parfaitement visible. Pour chaque question, il y a au moins une bonne réponse. Les copies seront scannées, il est donc fortement recommandé d'utiliser un stylo plutôt qu'un crayon pour éviter que certaines cases soient mal reconnues.

L'espérance de chaque question est de 0 : chaque bonne réponse rapporte $1/NB$ et chaque mauvaise réponse rapporte $-1/NM$ où NB est le nombre de bonnes réponses possibles et NM est le nombre de mauvaises réponses possibles. Vous pouvez donc obtenir un score négatif à une question.

QCM PS6

Question 1 Que réalise la route REST API suivante : POST /books/:id/authors/:authorId/roles ?

- ☐ Ajoute un rôle à un auteur spécifique dans un livre
- ☐ Ajoute un nouveau livre à la base de données
- ☐ Supprime un livre de la base de données
- ☐ Met à jour les informations d'un auteur spécifique associé à un livre

Question 2 Parmi les routes suivantes, laquelle respecte le principe de conception REST pour une API ?

- ☐ DELETE /books/deleteBook/123
- ☐ POST /books/createBook
- ☐ PUT /books/123/delete
- ☐ GET /books/updateBookTitle?bookId=123&newTitle=UpdatedTitle



Question 3 Quel code permet de créer une liste contenant chaque élément d'une liste myList ?

- ☐ `<ngFor="let item of myList" tag="ul">{{item}}</ngFor>`
- ☐ `<ul *ngFor="let item of myList">{{item}}`
- ☐ `<ul directive="ngFor" value="let item of myList">{{item}}`
- ☐ `<*ngFor="let item of myList">{{item}}</*ngFor>`

Question 4 Comment utiliser un service dans un composant Angular (en plus de l'importer) ?

- ☐ Il faut l'ajouter en paramètre du constructeur
- ☐ Il faut l'instancier dans ngOnInit()
- ☐ Rien, l'importer suffit
- ☐ Il faut le donner en paramètre des fonctions souhaitant l'utiliser

Question 5 Comment déclare-t-on un nouveau routeur pour gérer les questions liées à un quiz dans le fichier index.js d'une application Express Node.js ?

- ☐ `router.define('/:quizId/questions', QuestionsRouter)`
- ☐ `router.import('/:quizId/questions', QuestionsRouter)`
- ☐ `router.use('/:quizId/questions', QuestionsRouter)`
- ☐ `router.get('/:quizId/question', QuestionsRouter, (req, res) => { ... })`

Question 6 Quel code permet de créer un observable (qu'on appellera myObservable) sur une liste de Quiz en Angular ?

- ☐ `public myObservable:SubjectBehavior<Quiz[]> = new SubjectBehavior([])`
- ☐ `public myObservable:BehaviorSubject<Quiz[]> = new BehaviorSubject([])`
- ☐ `myObservable:Quiz[] = this.quizService.quizzes$.observe((quizzes: Quiz[]) => {})`
- ☐ `myObservable:Quiz[] = this.quizService.quizzes$.subscribe((quizzes: Quiz[]) => {})`

Question 7 À quoi sert un Input en Angular ?

- ☐ À envoyer une donnée d'un composant parent vers un composant enfant
- ☐ À afficher sur la page une "popup" dans laquelle l'utilisateur peut entrer une valeur
- ☐ À envoyer une donnée d'un composant enfant vers son composant parent
- ☐ À afficher dans la console un prompt dans lequel l'utilisateur peut entrer une valeur



Question 8 Quel code de réponse HTTP permet d'indiquer qu'une ressource a bien été créée ?

- ☐ 500
- ☐ 201
- ☐ 404
- ☐ 301

Question 9 Lors de la définition du modèle suivant, que signifie le "?" ?:

```
{  
id: string;  
questions?: Question[];  
}
```

- ☐ Qu'il doit absolument y avoir des questions
- ☐ Qu'il doit y avoir exactement 0 ou 1 question
- ☐ Qu'il peut ne pas y avoir de questions
- ☐ Rien, on ne peut pas mettre de ? lorsqu'on définit un modèle

Question 10 Quel code permet d'être notifié lorsque la valeur de l'observable présent dans `this.quizService.myObservable` change ?

- ☐ `public myObserver: SubjectBehavior<Quiz[]> = new SubjectBehavior(this.quizService.myObservable)`
- ☐ `this.quizService.myObservable.subscribe((quizzes: Quiz[]) => {})`
- ☐ `this.quizService.myObservable.observe((quizzes: Quiz[]) => {})`
- ☐ `public myObserver: BehaviorSubject<Quiz[]> = new BehaviorSubject(this.quizService.myObservable)`

Question 11 Comment exporter un modèle dans Angular ?

- ☐ `export interface Quiz {...}`
- ☐ `export model Quiz {...}`
- ☐ `export define Quiz {...}`
- ☐ `export Quiz: Model {...}`

Question 12 À quoi sert un Output en Angular ?

- ☐ À envoyer une donnée d'un composant enfant vers son composant parent
- ☐ À afficher une "popup" contenant un message donné sur la page
- ☐ À afficher un message dans la console
- ☐ À envoyer une donnée d'un composant parent vers un composant enfant



Question 13 Comment utiliser un modèle dans un service Angular (en plus de l'importer) ?

- ☐ Il faut le donner en paramètre des fonctions souhaitant l'utiliser
- ☐ Il faut l'instancier dans ngOnInit()
- ☐ Rien, l'importer suffit
- ☐ Il faut l'ajouter en paramètre du constructeur

Question 14 D'après les principes REST, que permet probablement de faire la requête GET /books/123/authors/ ?

- ☐ Récupérer la liste des auteurs du livre ayant pour id 123
- ☐ Modifier un auteur du livre d id 123
- ☐ Récupérer le livre ayant l'id 123
- ☐ Ajouter un auteur au livre d'id 123

Question 15 À quoi sert le bloc try...catch dans les différentes routes de l'API ?

- ☐ À définir les différentes routes disponibles dans l'application
- ☐ À gérer les erreurs potentielles lors de l'exécution du code à l'intérieur du bloc try
- ☐ À retourner une réponse JSON avec le statut de succès dans les routes correspondantes
- ☐ À effectuer des opérations de manipulation de données dans la base de données

Question 16 Quel est le rôle principal de la méthode HTTP GET dans les API REST ?

- ☐ Mettre à jour une ressource existante
- ☐ Récupérer des données d'une ressource
- ☐ Supprimer une ressource
- ☐ Créer une nouvelle ressource

Question 17 Pourquoi faut-il renvoyer un code d'état correct pour chaque réponse d'une API REST ?

- ☐ Cela réduit la complexité de l'implémentation de l'API
- ☐ Cela améliore les performances de l'API
- ☐ Cela permet aux clients d'interpréter correctement la réponse
- ☐ Cela garantit la sécurité de l'API



Question 18 Quel code permet d'afficher la propriété `myProp` de l'objet `myObj` (défini dans la partie ts du même composant) dans l'HTML du composant ?

- ☐ `<p><ngIf="myObj">{{myObj.myProp}}</ngIf></p>`
- ☐ `<p><if="myObj">{{myObj.myProp}}</if></p>`
- ☐ `<p *ngIf="myObj">{{myObj.myProp}}</p>`
- ☐ `<p *if="myObj">{{myObj.myProp}}</p>`

Question 19 Quel décorateur permet de définir une classe comme étant un service dans Angular ?

- ☐ `@Injectable`
- ☐ `@Service`
- ☐ `@Component`
- ☐ `@Singleton`

Question 20 Dans le contexte du fichier `index.js` de l'API Node.js, comment retourne-t-on un code de statut HTTP spécifique dans une route ?

- ☐ En utilisant la méthode `.setStatusCode()` de l'objet `res` et en lui passant le code de statut en paramètre
- ☐ En utilisant la méthode `.code()` de l'objet `res` et en lui passant le code de statut en paramètre
- ☐ En utilisant la méthode `.status()` de l'objet `res` et en lui passant le code de statut en paramètre
- ☐ En utilisant la méthode `.sendStatus()` de l'objet `res` et en lui passant le code de statut en paramètre

Question 21 Comment retourne-t-on une donnée `data` au format JSON dans une route du fichier `index.js` de l'API Node.js avec Express ?

- ☐ En utilisant la méthode `.status(data)`
- ☐ En utilisant la méthode `.json(data)`
- ☐ En utilisant la méthode `.data('json', data)`
- ☐ En utilisant la méthode `.send('json', data)`



Question 22 Puisqu'on n'utilise pas de vraie base de données dans le back-end de notre application, la description des données est faite grâce au package joi. Comment définir un string qui doit obligatoirement être passé ?

- ☐ {
 myProperty: Joi.string(true)
}
- ☐ {
 myProperty: Joi.required("string")
}
- ☐ {
 myProperty: Joi.string(false)
}
- ☒ {
 myProperty: Joi.string().required
}

Question 23 À quoi sert un service dans Angular ?

- ☐ À configurer l'aspect visuel d'une donnée
- ☐ À séparer la récupération des données de leur affichage
- ☐ À définir le modèle d'une donnée
- ☐ À assurer l'unicité des données dont il s'occupe

Question 24 Quelle classe faut-il utiliser pour permettre à un Output de transmettre sa valeur ?

- ☐ Output
- ☐ BehaviorSubject
- ☐ Aucune, il suffit d'utiliser @Output
- ☒ EventEmitter

Question 25 Comment récupère-t-on tous les quizzes dans la route GET / du fichier index.js ?

- ☐ En utilisant Quiz.retrieve()
- ☐ En utilisant Quiz.getById(req.params.quizId)
- ☐ En utilisant Quiz.getAll()
- ☒ En utilisant Quiz.get()



Question 26 Comment doit-on généralement envoyer les données lors d'une requête POST dans une API REST ?

- ☐ Par le biais des paramètres de requête (query parameters)
- ☐ Dans le corps de la requête (request body)
- ☐ Directement dans la route de l'URL, par exemple /books/123/names/nomDuLivreAAjouter
- ☐ Par l'utilisation d'un en-tête spécial de requête

Question 27 Comment utiliser un Input quiz dans un composant Angular (html) ?

- ☐ `<myComponent [quiz]="myQuiz"></myComponent>`
- ☐ `{quiz}`
- ☐ `<myComponent (quiz)="myQuiz"></myComponent>`
- ☐ `{{quiz}}`

Question 28 D'après les principes REST, quelle est la méthode la plus adaptée pour mettre à jour une information ?

- ☐ PUT
- ☐ POST
- ☐ GET
- ☐ DELETE

Question 29 Comment déclarer qu'on attend un Output quiz dans un composant Angular (html) ?

- ☐ `<myComponent [quiz]="callBackFunction(event)"></myComponent>`
- ☐ `{quiz}`
- ☐ `<myComponent (quiz)="callBackFunction(event)"></myComponent>`
- ☐ `{{quiz}}`

Question 30 Quel code permet à l'observable `this.myObservable` d'indiquer le changement de la valeur qu'il permet d'observer ?

- ☐ `this.myObservable.value(newValue)`
- ☐ `this.myObservable.next(newValue)`
- ☐ `this.myObservable.set(newValue)`
- ☐ `this.myObservable = newValue`



Question 31 Comment déclarer un Input dans un composant Angular (ts) ?

- ☐ En ajoutant Input aux paramètres du constructeur
- ☐ Avec console.log
- ☐ En le déclarant comme type des variables qui feront office d'Input
- ☐ Grâce au décorateur @Input

Question 32 Comment récupère-t-on l'identifiant d'un quiz à partir des paramètres de la requête dans la route /:quizId du fichier index.js de l'API ?

- ☐ En utilisant req.headers.quizId
- ☐ En utilisant req.body.quizId
- ☐ En utilisant req.params.quizId
- ☐ En utilisant req.query.quizId