

Aide-mémoire : Node.js et Express.js



Voici les objets et méthodes que vous devrez connaître pour MIS3502. Dans de nombreux cas, les notes et les exemples ont été simplifiés. Pour une documentation complète, veuillez consulter : <https://expressjs.com/en/4x/api.html> et <https://nodejs.org/en/docs/>

Objet et/ou méthode	Nom de la variable (par convention)	Remarques
L'objet <code>express</code> .	<code>express</code>	Objet de niveau supérieur. Lorsque la méthode <code>express()</code> est appelée, elle renvoie un objet d'application <code>express</code> . Vous avez besoin de cet objet d'application pour utiliser toutes les autres fonctionnalités d' <code>Express.js</code> .
L'objet analyseur de corps	<code>bodyParser</code>	<p>L'objet <code>body-parser</code> contient une variété de méthodes qui renvoient les fonctions. (Vous avez bien lu – c'est une méthode qui renvoie une fonction !)</p> <p>Le but de ces fonctions est de fournir des instructions sur la façon de coder les données entrant dans la fonction.</p> <p>Ainsi, si les données entrantes doivent être codées en URL, vous devez utiliser <code>bodyParser.urlencoded()</code>. Si les données entrantes devraient être JSON, vous pouvez utiliser <code>bodyParser.json()</code>.</p> <p>Dans notre classe, les données entrantes seront toujours codées en URL et les données sortantes de l'API seront toujours en JSON.</p> <p>N'oubliez pas : URL codée en entrée, sortie JSON.</p>
<code>bodyParser.urlencoded({extended:false})</code>	N'est pas applicable	<p>La méthode codée en URL de l'objet <code>bodyParser</code> renvoie un fonction avec des instructions sur la façon de gérer les données codées en URL entrantes. Il doit être fourni avec un paramètre <code>{extended:false}</code> ou <code>{extended:true}</code>.</p> <p>Dans notre classe, nous utiliserons toujours <code>{extended:false}</code>.</p> <p>L'option « étendue » est liée à la complexité attendue des données codées dans l'URL entrante. Sera-t-il multidimensionnel (étendu) ou non ?</p> <p>Les instructions incluent des appels à la fonction <code>next()</code> afin que l'exécution du code ne se termine pas prématurément.</p>

Objet et/ou méthode	Nom de variable (par convention)	Remarques
L'objet application express	application	<p>L'objet app est l'objet le plus important fourni par le Cadre express. Il propose un certain nombre de méthodes utiles :</p> <ul style="list-style-type: none"> • app.use • app.get • app.post • app.delete • app.put • app.listen
La méthode d'utilisation de l'objet app	app.use()	<p>L'application Express appellera la fonction de rappel sans tenir compte du chemin ou de la méthode (POST ou GET) de la requête entrante.</p> <p>Syntaxe de base :</p> <pre>app.use(function(req,res,suivant){ //le code va ici });</pre>
La méthode GET de l'application objet	app.get()	<p>Interceptez et gérez un événement HTTP GET.</p> <p>La méthode get() prend un chemin comme argument, ainsi qu'une fonction de rappel. La fonction de rappel fournit une requête et un objet de réponse.</p> <p>Le chemin peut être « / », ce qui signifie qu'il n'y a pas de chemin.</p> <p>Le chemin peut être « /xyz » qui intercepterait un HTTP POST ciblé sur xyz. La valeur 'xyz' est virtuelle. Il ne correspond pas à un dossier sur le système de fichiers du serveur.</p> <p>Syntaxe de base :</p> <pre>app.get('/xyz',function(req,res){ //le code va ici });</pre>
La méthode POST du objet d'application	app.post()	Interceptez et gérez un événement HTTP POST. Similaire à app.get.
La méthode DELETE du objet d'application	app.delete()	Interceptez et gérez un événement HTTP DELETE. Similaire à app.get.

Objet et/ou méthode	Nom de variable (par convention)	Remarques
La méthode PUT de l'application objet	app.put()	Piège et gère un événement HTTP PUT. Similaire à app.get.
La méthode d'écoute du objet d'application	app.écouter()	<p>Cette méthode définit le port que l'objet d'application express écoutera sur. La fonction de rappel s'exécute lorsque l'application a été démarrée. La méthode renvoie un objet qui résume les propriétés du serveur.</p> <p>//ici XXXX est le numéro de port app.listen(XXXX,function() { });</p>
L'objet de requête express req		<p>Toutes les fonctions de rappel de l'application fournissent un objet de requête. La variable « req » est l'abréviation de « request ». Il représente les données envoyées au point de terminaison de l'API. Le framework Express (qui inclut un analyseur de corps) améliore et simplifie l'objet de requête fourni par Node.js seul.</p> <p>L'objet de requête aura des objets enfants de requête et de corps, qui à leur tour auront des propriétés qui correspondent aux données envoyées via GET et POST.</p> <p>Par exemple:</p> <p>req.query.x // fait référence à une requête // paramètre de chaîne x</p> <p>corps requis.y // fait référence à une balise de formulaire // portant le nom de y</p>
La réponse expresse objet	rés	<p>Toutes les fonctions de rappel de l'application fournissent un objet de réponse. La variable « res » est l'abréviation de « réponse ». Il représente les données envoyées depuis le point de terminaison de l'API. Le framework Express (qui inclut un analyseur de corps) améliore et simplifie l'objet de réponse fourni par Node.js seul.</p> <p>L'objet de réponse aura les méthodes suivantes : • header() • writeHead() • write() • end()</p>
La méthode d'en-tête du objet de réponse	res.header()	<p>Définit le champ d'en-tête HTTP de la réponse sur valeur. En-tête multiple Des paires clé-valeur peuvent être écrites. Cette méthode n'est en réalité qu'un alias pour res.set().</p>

Objet et/ou méthode	Nom de variable (par convention)	Remarques
La méthode writeHead de l'objet de réponse	res.writeHead()	Écrivez le code d'état HTTP (par exemple 200 = succès) et tous les en-têtes de réponse HTTP qui l'accompagnent. Cette méthode ne doit être appelée qu'une seule fois sur un message et elle doit être appelée avant l'appel de réponse.end().
La méthode d'écriture du objet de réponse	res.write()	Écrivez un flux de texte dans la réponse HTTP.
La méthode de fin du objet de réponse	res.end()	Terminez la réponse HTTP.
L'objet suivant express	suivant()	Les fonctions de rappel de l'application peuvent également fournir un objet suivant. Lorsque la méthode next() est appelée, le prochain événement d'application express correspondant sera évalué.

