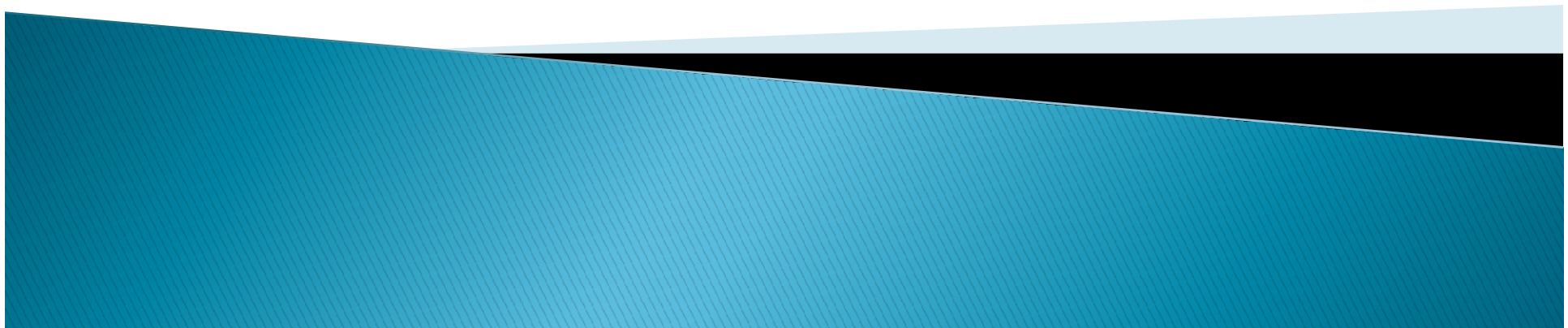


# พัฒนา Modern Web Application ด้วย Node.js

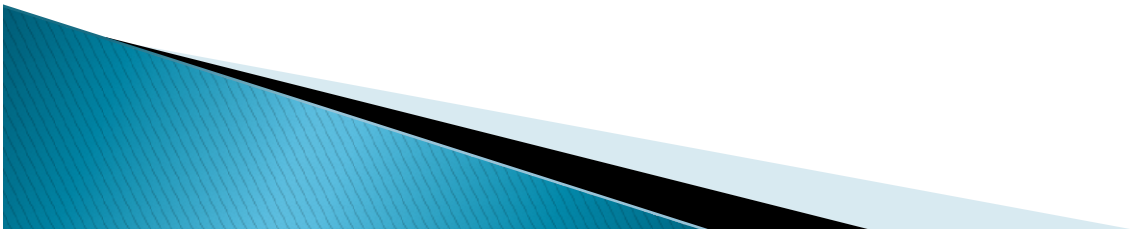
โค้ชเอก

[Codingthailand.com](http://Codingthailand.com)



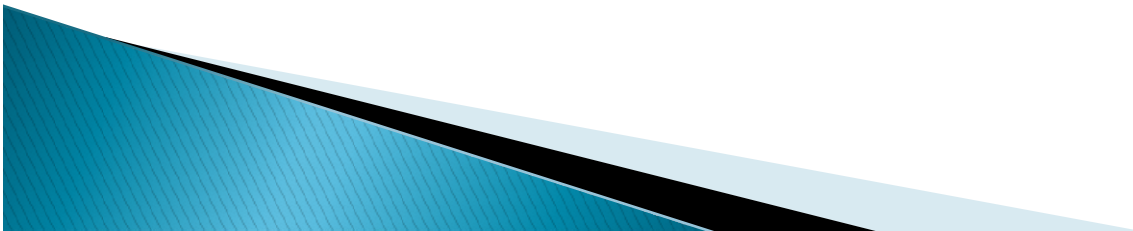
# Express.js : Routing

```
app.get("/", function(req, res) {  
  res.send("you just sent a GET request, friend");  
});  
  
app.post("/", function(req, res) {  
  res.send("a POST request? nice");  
});  
  
app.put("/", function(req, res) {  
  res.send("i don't see a lot of PUT requests anymore");  
});  
  
app.delete("/", function(req, res) {  
  res.send("oh my, a DELETE??");  
});  
  
app.listen(3000, function() {  
  console.log("App is listening on port 3000");  
});
```



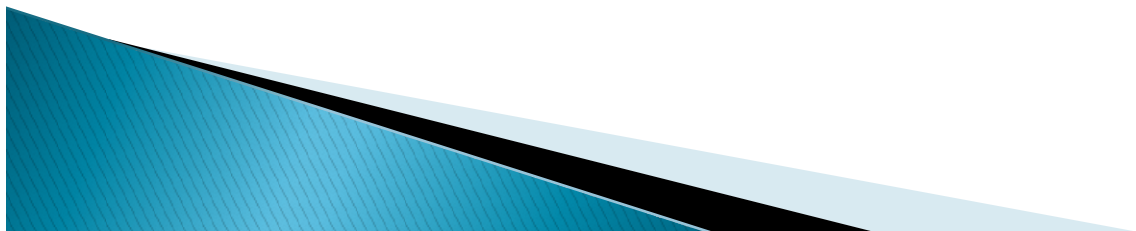
# Express.js

```
app.get('/books/:id', function (request, response) {  
  console.log('Book details from book id ' + request.params.id);  
});
```



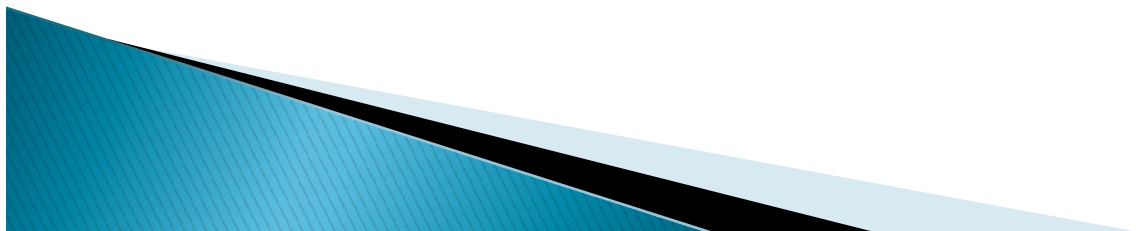
# Express.js : Request Object

- ▶ request.query: query string parameters
- ▶ request.params: URL parameters
- ▶ request.body: request body data
- ▶ request.route: the route path
- ▶ request.cookies: cookie data
- ▶ request.signedCookies: signed cookie data
- ▶ request.header() and request.get(): request headers
- ▶ ดูทั้งหมดได้ที่ <http://expressjs.com/en/api.html#req>



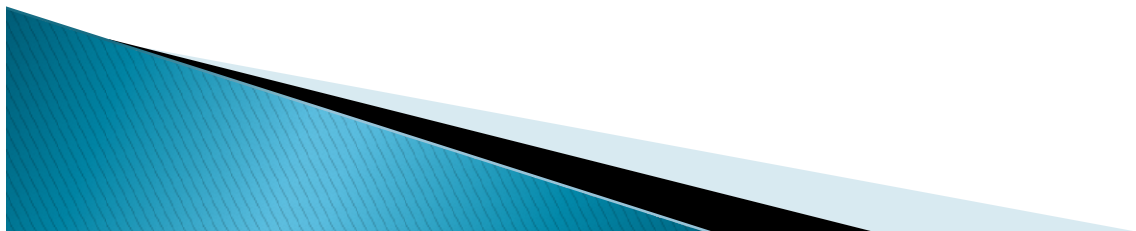
# Express.js : Request Object

- ▶ `request.query`
- ▶ ตัวอย่าง Url เช่น `http://localhost:3000/about?a=10&b=5`
- ▶ `let content = req.query;`
- ▶ `console.log(JSON.stringify(content));`
- ▶ ผลลัพธ์ คือ `{"a":"10","b":"5"}`



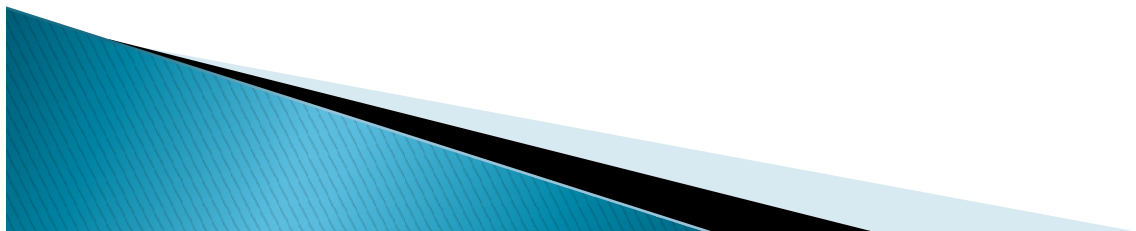
# Express.js : Request Object

- ▶ `request.body`
- ▶ `json()`: For parsing HTTP(S) payload into JavaScript/Node.js objects
- ▶ `urlencoded()`: For parsing URL-encoded HTTP(S) requests' data into JavaScript/Node.js objects



# Express.js : Response Object

- ▶ `response.render()`
- ▶ `response.locals`
- ▶ `response.set()`
- ▶ `response.status()`
- ▶ `response.send()`
- ▶ `response.json()`
- ▶ `response.jsonp()`
- ▶ `response.redirect()`
  
- ▶ <http://expressjs.com/en/api.html#res>



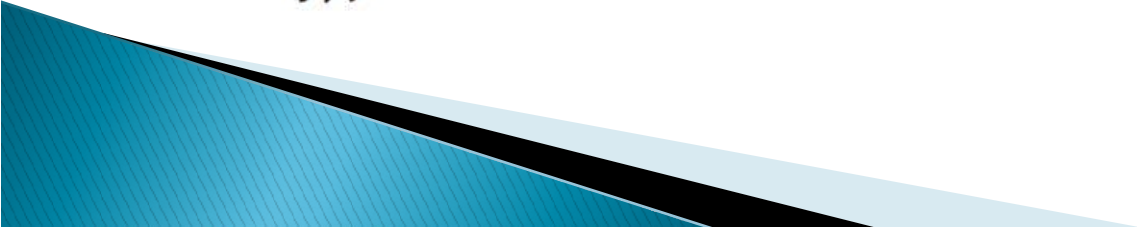
# Express.js : Response Object

- ▶ response.render()

```
app.get('/render-title', function(req, res) {  
  res.render('index', {title: 'Pro Express.js'});  
});
```

- ▶ response.locals

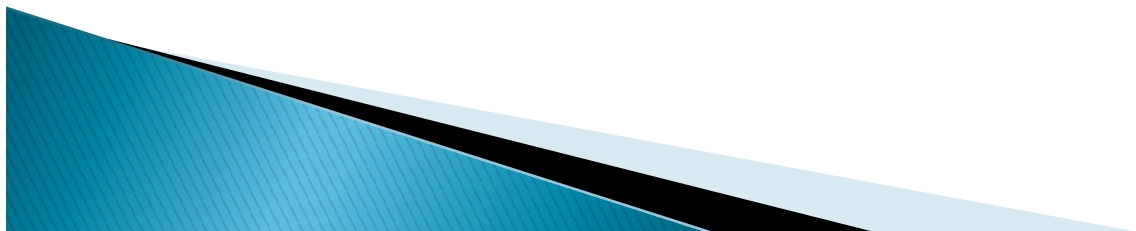
```
app.get('/locals', function(req, res){  
  res.locals = { title: 'Pro Express.js' };  
  res.render('index');  
});
```





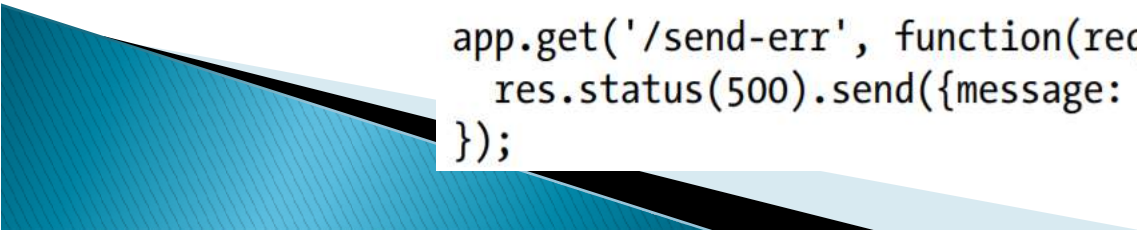
# Express.js : Response Object

- ▶ `response.status()`
- ▶ 200: OK
- ▶ 201: Created
- ▶ 301: Moved Permanently
- ▶ 401: Unauthorized
- ▶ 403: FORBIDDEN ERRORS
- ▶ 404: Not Found
- ▶ 500: Internal Server Error
- ▶ `res.status(404);`
- ▶ `res.status(404).json({ error: "Resource not found!" });`



# Express.js : Response Object

- ▶ `response.send()`
- ▶ **String:** `response.send('success');` with `text/html`
- ▶ **Object:** `response.send({message: 'success'});`  
or `response.send({message: 'error'});` with JSON representation
- ▶ **Array:** `response.send([{'title: 'Practical Node.js'}, {'title: 'Rapid Prototyping with JS'}]);` with JSON representation
- ▶ **Buffer:** `response.send(new Buffer('Express.js Guide'));` with `application/octetstream`

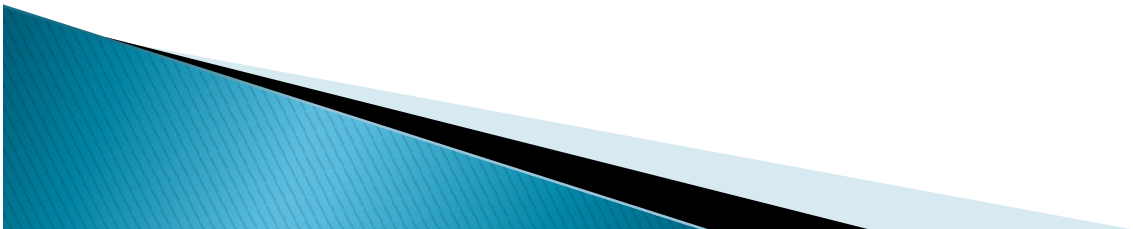


```
app.get('/send-err', function(req, res) {  
  res.status(500).send({message: 'Oops, the server is down.'});  
});
```

# Express.js : Response Object

- ▶ `response.json()`
- ▶ จะกำหนด Content-Type เป็น `application/json` ให้อย่างแน่นอนตัวนี้จะได้ไม่ต้องเขียนโค้ดเยอะ

```
app.get('/json', function(req, res) {  
  res.status(200).json([  
    {title: 'Practical Node.js', tags: 'node.js express.js'},  
    {title: 'Rapid Prototyping with JS', tags: 'backbone.js node.js mongodb'},  
    {title: 'JavaScript: The Good Parts', tags: 'javascript'}  
  ]);  
});
```

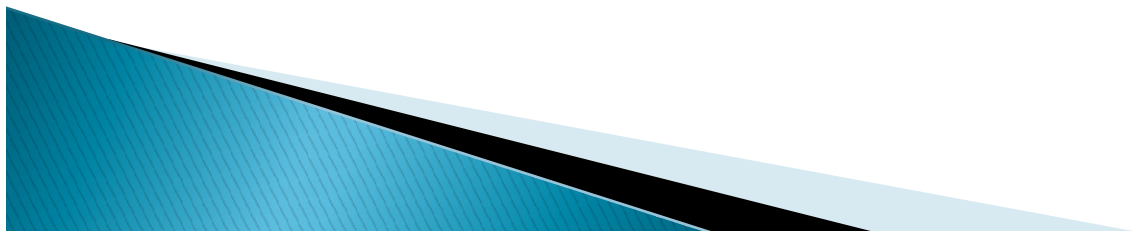


# Express.js : Response Object

## ▶ response.redirect()

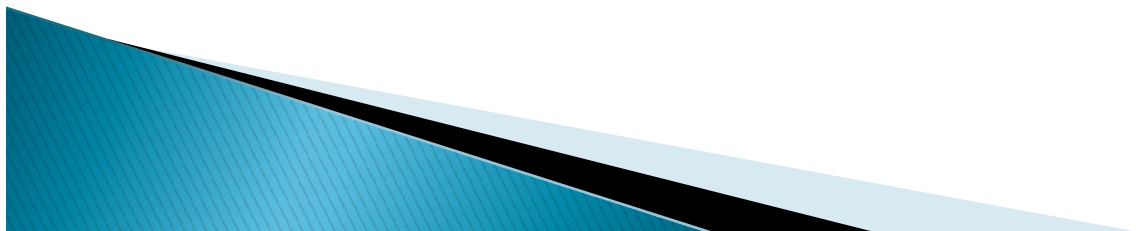
```
res.redirect('/admin');  
res.redirect('../users');  
res.redirect('http://rapidprototypingwithjs.com');
```

```
res.redirect(301, 'http://rpjs.co');
```



# Express.js

- ▶ ทำแบบฝึกหัดสร้าง Blog



The end

