

1. Create a new route in app.js

```
2. app.use(require('./routes/feedback'));
3. app.use(require('./routes/api'))
```

4. routes/feedback.js

```
var express = require('express');
var router = express.Router();

router.get('/feedback', function(req, res) {

  res.render('feedback', {
    pageTitle: 'Feedback',
    pageID: 'feedback'
  });

});

module.exports = router;
```

5. Show views/feedback.ejs

6. Public/js/feedback.js

```
7. $(function() {
8.   $.getJSON('api', updateFeedback);
9.
10.  $('#feedback-form').submit(function(e) {
11.    e.preventDefault();
12.    $.post('api', {
13.      name: $('#feedback-form-name').val(),
14.      title: $('#feedback-form-title').val(),
15.      message: $('#feedback-form-message').val()
16.    }, updateFeedback);
17.  });
18.
19.  $('#feedback-messages').on('click', function(e) {
20.    if (e.target.className == 'glyphicon glyphicon-remove') {
21.      $.ajax({
22.        url: 'api/' + e.target.id,
23.        type: 'DELETE',
24.        success: updateFeedback
25.      }); //ajax
26.    } // the target is a delete button
27.  }); //feedback messages
```

28. Routes/api

```
var express = require('express');
```

```

var router = express.Router();
var bodyParser = require('body-parser');
var fs = require('fs');
var feedbackData = require('../data/feedback.json');

router.get('/api', function(req, res) {
  res.json(feedbackData);
});

router.use(bodyParser.json());
router.use(bodyParser.urlencoded({ extended: false }));

router.post('/api', function(req, res) {
  feedbackData.unshift(req.body);
  fs.writeFile('app/data/feedback.json', JSON.stringify(feedbackData), 'utf8',
function(err) {
  if (err) {
    console.log(err);
  }
});
  res.json(feedbackData);
});

router.delete('/api/:id', function(req, res) {
  feedbackData.splice(req.params.id, 1);
  fs.writeFile('app/data/feedback.json', JSON.stringify(feedbackData), 'utf8',
function(err) {
  if (err) {
    console.log(err);
  }
});
  res.json(feedbackData);
});

module.exports = router;

```

29. Lsdk

30. Sldkf

31. sldkf