

# How to use **post** function in Express

## Best JavaScript code snippets using express.Express.post

(Showing top 15 results out of 7,731)

*Write less, code more.*

AI Code Completion  
Plugin For Your IDE

Get Tabnine

```
function calculateRunTime() {  
  const startTime = new Date();  
  doSomething();  
  const endTime = new Date();  
  let di  
}
```

```
diff = endTime - startTime;  
diff = endTime() - startTime();  
diff = startTime - endTime;  
diff =
```

src/server.js/app.post

origin: [schn4ck/s...](#)

```
// for markdown preview  
app.post('/markdown', (request, reply) => {  
  const { comment } = request.body;  
  reply.send({ html: marked(comment.trim()) });  
});
```

server.js/api.post

origin: [lando/lan...](#)

```
/*  
 * Post new meta data for metrics.  
 */  
api.post('/metrics/v2/:id', handler((req, res) => {  
  return Promise.map(plUGINS, plugin => {  
    const reporter = new plugin.Reporter(config[plugin.config])  
    return reporter.ping()  
  }).then(() => reporter.report(_.merge({}, req.body, {inst
```

```
.then(() => log.info('Reported to %s', plugin.name))
.then(() => reporter.close());
});
}));
```

## src/server.js/app.post

origin: schn4ck/s...

```
// settings
app.post('/setting/:property/:value', (request, reply) => {
  const { property, value } = request.params;
  const user = getUser(request);
  if (!isAdmin(user)) return reply.status(403).send('Access denied');
  const setting = value ? 1 : 0;
  db.run(queries.set_settings, [property, setting], err => {
    if (error(err, request, reply)) return;
    reply.send({ status: 'ok' });
  });
});
```

## examples/receiving-api.js/api.post

origin: esbenp/pd...

```
api.post('/hook', function (req, res) {
  var signature = req.get('X-PDF-Signature', 'sha1=')

  var bodyCrypted = require('crypto')
    .createHmac('sha1', '12345')
    .update(JSON.stringify(req.body))
    .digest('hex')

  if (bodyCrypted !== signature) {
    res.status(401).send()
    return
  }
})
```

```
console.log('PDF webhook received', JSON.stringify(req.bo  
  
res.status(204).send()  
}))
```

spec/HttpRequest.spec.js/startSe... *origin: parse-com...*

```
function startServer(done) {  
  const app = express();  
  app.use(bodyParser.json({ type: '*/*' }));  
  app.get('/hello', function(req, res) {  
    res.json({ response: 'OK' });  
  });  
  
  app.get('/404', function(req, res) {  
    res.status(404);  
    res.send('NO');  
  });  
  
  app.get('/301', function(req, res) {  
    res.status(301);  
    res.location('/hello');  
    res.send();  
  });  
  
  app.post('/echo', function(req, res) {  
    res.json(req.body);  
  });  
  
  app.get('/qs', function(req, res) {  
    res.json(req.query);  
  });
```

```
return app.listen(13371, undefined, done);  
}
```

## src/server.js/app.post

origin: schn4ck/s...

```
// trust/block users or approve/reject comments  
app.post(  
  /\/(? :comment\/(\d+)\/(approve|reject))|(? :user\/(\d+)\d+)(?  
(request, reply) => {  
  const user = getUser(request);  
  if (!isAdmin(user)) return reply.status(403).send(re  
  const action = request.params[1] || request.params[3  
  const target_id = +(request.params[0] || request.par  
  db.run(queries[action], target_id, err => {  
    if (error(err, request, reply)) return;  
    reply.send({ status: 'ok' });  
  });  
});  
);
```

## src/api.js/createApi

origin: esbenp/pd...

```
api.post('/', function(req, res) {  
  var queue = createQueue()  
  var authHeader = req.get('Authorization')
```

## src/server.js/app.post

origin: schn4ck/s...

```
// POST new comment  
app.post('/comments/:slug', (request, reply) => {  
  const { slug } = request.params;  
  const { comment, replyTo } = request.body;  
  const user = getUser(request);
```

```
if (!user) return error('access denied', request, repl
checkValidComment(db, slug, user.id, comment, replyTo,
  if (err) return reply.send({ status: 'rejected', rea
let stmt = db
  .prepare(queries.insert, [user.id, slug, comment,
  .run(err => {
    if (err) return error(err, request, reply);
    if (!user.blocked && !user.trusted) {
      awaiting_moderation.push({ slug });
    }
    schnackEvents.emit('new-comment', {
      user: user,
      slug,
      id: stmt.lastID,
      comment,
      replyTo
    });
    reply.send({ status: 'ok', id: stmt.lastID });
  });
});
});
```

server.js/app.post

origin: GoogleClo...

```
app.post('/', async (req, res, next) => {
  const {team} = req.body;
  const timestamp = new Date();
```

lib/httpServer/index.js/app.post

origin: conradoqg...

```
this.app.post('/miner/mine', (req, res, next) => {
  miner.mine(req.body.rewardAddress, req.body['feeAddr
```

```
.then((newBlock) => {  
  newBlock = Block.fromJson(newBlock);  
  blockchain.addBlock(newBlock);  
  res.status(201).send(newBlock);  
})  
.catch((ex) => {  
  if (ex instanceof BlockAssertionError && ex.mess  
  else next(ex);  
});  
});
```

lib/httpServer/index.js/app.post

origin: conradoqg...

```
this.app.post('/blockchain/transactions', (req, res) => {  
  let requestTransaction = Transaction.fromJson(req.bo  
  let transactionFound = blockchain.getTransactionById  
  
  if (transactionFound != null) throw new HTTPError(40  
  
  try {  
    let newTransaction = blockchain.addTransaction(req  
    res.status(201).send(newTransaction);  
  } catch (ex) {  
    if (ex instanceof TransactionAssertionError) throw  
    else throw ex;  
  }  
});
```

src/app.js/app.post

origin: dialogflo...

```
app.post('/sms', (req, res) => {  
  
  console.log('POST sms received');
```

```
try {
  bot.processMessage(req, res);
} catch (err) {
  return res.status(400).send('Error while processing ');
}
});
```

lib/httpServer/index.js/app.post

origin: conradoqg...

```
this.app.post('/operator/wallets', (req, res) => {
  let password = req.body.password;
  if (R.match(/\w+/g, password).length <= 4) throw new

  let newWallet = operator.createWalletFromPassword(pa

  let projectedWallet = projectWallet(newWallet);

  res.status(201).send(projectedWallet);
});
```

metrics/server.js/api.post

origin: lando/lan...

```
/*
 * Post new meta data for metrics.
 */
api.post('/metrics/v2/:id', handler((req, res) => {
  return Promise.map(plugings, plugin => {
    const reporter = new plugin.Reporter(config[plugin.confir
    return reporter.ping()
      .then(() => reporter.report(_.merge({}, req.body, {inst
      .then(() => log.info('Reported to %s', plugin.name))
      .then(() => reporter.close()));
```

```
});  
}});
```

src/app.js/app.post

origin: dialogflo...

```
app.post('/webhook', (req, res) => {  
  console.log('POST webhook');  
  
  try {  
    bot.processMessage(req, res);  
  } catch (err) {  
    return res.status(400).send('Error while processing '  
  }  
});
```