|  |  |
| --- | --- |
| TOPIC/TITLE  level 4 Salting and Hashing Password with bcrypt | |
| Keywords/Questions  Salting  Bcrypt  Salt rounds  Bcrypt insatalation  App.post.register  App.post.login | Notes : |
|  |
|  |
|  |
| Increase randomly the number of character before generating the hash conversion |
|  |
|  |
|  |
| So it will turn that if people use the same password they won’t have the same hash code |
| Its tored in the db along with the hash |
|  |
| This is one of the industry standard hashing algorithmas that developpers use to keep their users passwords in order to slow the decryption process |
|  |
|  |
| We can salt after once hashed compilation |
| With bcrypt you can salt with how many rounds you want |
|  |
| https://www.npmjs.com/package/bcrypt |
|  |
| You have to install an older version of node using in your HT:  curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.37.2/install.sh | bash  Nvm install 14.15.4 |
| Npm I bcrypt |
| const bcrypt = require ("bcrypt");  const saltRounds = 10; |
|  |
| Then in our app.post register |
|  |
| bcrypt.hash(myPlaintextPassword, salt, function(err, hash) {  *// Store hash in your password DB.*      });  app.post("/register", function(req,res){  bcrypt.hash(req.body.password, saltRounds, function(err, hash) {  const newUser = new User ({// don't forget to capitalise User  email:req.body.username,  password: hash  });  newUser.save(function(err){  if(err){  console.log(err);  }else{  res.render("secrets")  }  }) // Store hash in your password DB.  });  }); |
|  |
| *// Load hash from your password DB.*  bcrypt.compare(myPlaintextPassword, hash, function(err, result) {  *// result == true*  }); |
| app.post("/login",function(req,res){  const username = req.body.username;  const password = req.body.password;  User.findOne({email:username}, function(err, foundUser){  if (err) {  console.log(err);  }else{  if(foundUser){  // Load hash from your password DB.  bcrypt.compare(password, foundUser.paswword, function(err, result) {  if(result===true){  res.render("secrets");  }  });  };  };  });  }); |
|  |
|  |
|  |
|  |
|  |
|  |
|  | |