

EventAdmin

1.0

Generated by Doxygen 1.14.0

1 File Index	1
1.1 File List	1
2 File Documentation	3
2.1 src/config/db.js File Reference	3
2.1.1 Variable Documentation	3
2.1.1.1 connectDB	3
2.1.1.2 exports	3
2.1.1.3 mongoose	3
2.2 src/controllers/authController.js File Reference	4
2.2.1 Variable Documentation	4
2.2.1.1 bcrypt	4
2.2.1.2 const	4
2.2.1.3 crypto	4
2.2.1.4 forgotPassword	4
2.2.1.5 jwt	4
2.2.1.6 login	5
2.2.1.7 resetPassword	5
2.2.1.8 resetTokens	5
2.2.1.9 User	5
2.3 src/controllers/organizerController.js File Reference	5
2.3.1 Variable Documentation	6
2.3.1.1 bcrypt	6
2.3.1.2 const	6
2.3.1.3 createOrganizer	6
2.3.1.4 deleteOrganizer	6
2.3.1.5 getAllOrganizers	6
2.3.1.6 Organizer	6
2.3.1.7 Profile	6
2.3.1.8 updateOrganizer	6
2.3.1.9 User	7
2.4 src/index.js File Reference	7
2.4.1 Variable Documentation	7
2.4.1.1 app	7
2.4.1.2 authRoutes	7
2.4.1.3 connectDB	7
2.4.1.4 cors	7
2.4.1.5 express	7
2.4.1.6 organizerRoutes	7
2.4.1.7 PORT	8
2.5 src/middlewares/authMiddleware.js File Reference	8
2.5.1 Variable Documentation	8

2.5.1.1 exports	8
2.5.1.2 jwt	8
2.6 src/models/Level.js File Reference	8
2.6.1 Variable Documentation	8
2.6.1.1 exports	8
2.6.1.2 LevelSchema	9
2.6.1.3 mongoose	9
2.7 src/models/Organizer.js File Reference	9
2.7.1 Variable Documentation	9
2.7.1.1 exports	9
2.7.1.2 mongoose	9
2.7.1.3 OrganizerSchema	9
2.8 src/models/Profile.js File Reference	10
2.8.1 Variable Documentation	10
2.8.1.1 exports	10
2.8.1.2 mongoose	10
2.8.1.3 ProfileSchema	10
2.9 src/models/User.js File Reference	10
2.9.1 Variable Documentation	10
2.9.1.1 exports	10
2.9.1.2 mongoose	11
2.9.1.3 UserSchema	11
2.10 src/routes/authRoutes.js File Reference	11
2.10.1 Variable Documentation	11
2.10.1.1 const	11
2.10.1.2 exports	11
2.10.1.3 router	11
2.11 src/routes/organizerRoutes.js File Reference	12
2.11.1 Variable Documentation	12
2.11.1.1 auth	12
2.11.1.2 const	12
2.11.1.3 exports	12
2.11.1.4 router	12
2.12 src/utls/emailService.js File Reference	12
2.12.1 Function Documentation	13
2.12.1.1 if()	13
2.12.1.2 sendWelcomeEmail()	13
2.12.2 Variable Documentation	13
2.12.2.1 apiKey	13
2.12.2.2 else	13
2.12.2.3 exports	13
2.12.2.4 sgMail	13

Chapter 1

File Index

1.1 File List

Here is a list of all files with brief descriptions:

src/ index.js	7
src/config/ db.js	3
src/controllers/ authController.js	4
src/controllers/ organizerController.js	5
src/middlewares/ authMiddleware.js	8
src/models/ Level.js	8
src/models/ Organizer.js	9
src/models/ Profile.js	10
src/models/ User.js	10
src/routes/ authRoutes.js	11
src/routes/ organizerRoutes.js	12
src/utls/ emailService.js	12

Chapter 2

File Documentation

2.1 src/config/db.js File Reference

Variables

- `const mongoose` = `require("mongoose")`
- `const connectDB`
- module `exports` = `connectDB`

2.1.1 Variable Documentation

2.1.1.1 connectDB

```
const connectDB
```

Initial value:

```
= async () => {  
  try {  
    await mongoose.connect(process.env.MONGO_URI, {  
      useNewUrlParser: true,  
      useUnifiedTopology: true,  
    });  
    console.log("MongoDB Bağlandı");  
  } catch (err) {  
    console.error(err);  
    process.exit(1);  
  }  
}
```

2.1.1.2 exports

```
module exports = connectDB
```

2.1.1.3 mongoose

```
const mongoose = require("mongoose")
```

2.2 src/controllers/authController.js File Reference

Variables

- `const bcrypt` = require("bcryptjs")
- `const jwt` = require("jsonwebtoken")
- `const crypto` = require("crypto")
- `const { sendWelcomeEmail }` = require("../utils/emailService")
- `const User` = require("../models/User")
- `const resetTokens` = {}
- `exports login`
- `exports forgotPassword`
- `exports resetPassword`

2.2.1 Variable Documentation

2.2.1.1 bcrypt

```
const bcrypt = require("bcryptjs")
```

2.2.1.2 const

```
const { sendWelcomeEmail } = require("../utils/emailService")
```

2.2.1.3 crypto

```
const crypto = require("crypto")
```

2.2.1.4 forgotPassword

```
exports forgotPassword
```

Initial value:

```
= async (req, res) => {  
  const { email } = req.body;  
  const user = await User.findOne({ email });  
  if (!user) return res.status(400).json({ msg: "Kullanıcı bulunamadı" });  
  const token = crypto.randomBytes(32).toString("hex");  
  resetTokens[token] = { userId: user._id, expires: Date.now() + 1000 * 60 * 15 };  
  
  console.log(`Şifre sıfırlama linki: http:  
  res.json({ msg: "Şifre sıfırlama bağlantısı e-posta ile gönderildi." });  
}
```

2.2.1.5 jwt

```
const jwt = require("jsonwebtoken")
```

2.2.1.6 login

```
exports login
```

Initial value:

```
= async (req, res) => {
  const { email, password } = req.body;
  const user = await User.findOne({ email });
  if (!user) return res.status(400).json({ msg: "Kullanıcı bulunamadı" });
  const isMatch = await bcrypt.compare(password, user.password);
  if (!isMatch) return res.status(400).json({ msg: "Şifre yanlış" });
  const token = jwt.sign(
    { id: user._id, role: user.role },
    process.env.JWT_SECRET,
    { expiresIn: "1h" }
  );
  res.json({ token });
}
```

2.2.1.7 resetPassword

```
exports resetPassword
```

Initial value:

```
= async (req, res) => {
  const { token, password } = req.body;
  const data = resetTokens[token];
  if (!data || data.expires < Date.now()) {
    return res.status(400).json({ msg: "Token geçersiz veya süresi dolmuş." });
  }
  const user = await User.findById(data.userId);
  if (!user) return res.status(400).json({ msg: "Kullanıcı bulunamadı." });
  user.password = await bcrypt.hash(password, 10);
  await user.save();
  delete resetTokens[token];
  res.json({ msg: "Şifre başarıyla güncellendi." });
}
```

2.2.1.8 resetTokens

```
const resetTokens = {}
```

2.2.1.9 User

```
const User = require("../models/User")
```

2.3 src/controllers/organizerController.js File Reference

Variables

- `const bcrypt` = `require("bcryptjs")`
- `const User` = `require("../models/User")`
- `const Organizer` = `require("../models/Organizer")`
- `const Profile` = `require("../models/Profile")`
- `const { sendWelcomeEmail }` = `require("../utils/emailService")`
- `exports createOrganizer`
- `exports getAllOrganizers`
- `exports updateOrganizer`
- `exports deleteOrganizer`

2.3.1 Variable Documentation

2.3.1.1 bcrypt

```
const bcrypt = require("bcryptjs")
```

2.3.1.2 const

```
const { sendWelcomeEmail } = require("../utils/emailService")
```

2.3.1.3 createOrganizer

```
exports createOrganizer
```

2.3.1.4 deleteOrganizer

```
exports deleteOrganizer
```

2.3.1.5 getAllOrganizers

```
exports getAllOrganizers
```

Initial value:

```
= async (req, res) => {  
  try {  
    const organizers = await Organizer.find().populate({  
      path: "user",  
      select: "email role",  
      populate: { path: "profile", select: "firstName lastName" },  
    });  
    return res.json(organizers);  
  } catch (err) {  
    console.error(err);  
    return res  
      .status(500)  
      .json({ msg: "Sunucu hatası: organizatörler alınamadı." });  
  }  
}
```

2.3.1.6 Organizer

```
const Organizer = require("../models/Organizer")
```

2.3.1.7 Profile

```
const Profile = require("../models/Profile")
```

2.3.1.8 updateOrganizer

```
exports updateOrganizer
```

2.3.1.9 User

```
const User = require("../models/User")
```

2.4 src/index.js File Reference

Variables

- `const express` = require("express")
- `const cors` = require("cors")
- `const connectDB` = require("../config/db")
- `const authRoutes` = require("../routes/authRoutes")
- `const organizerRoutes` = require("../routes/organizerRoutes")
- `const app` = express()
- `const PORT` = process.env.PORT || 5000

2.4.1 Variable Documentation

2.4.1.1 app

```
const app = express()
```

2.4.1.2 authRoutes

```
const authRoutes = require("../routes/authRoutes")
```

2.4.1.3 connectDB

```
const connectDB = require("../config/db")
```

2.4.1.4 cors

```
const cors = require("cors")
```

2.4.1.5 express

```
const express = require("express")
```

2.4.1.6 organizerRoutes

```
const organizerRoutes = require("../routes/organizerRoutes")
```

2.4.1.7 PORT

```
const PORT = process.env.PORT || 5000
```

2.5 src/middlewares/authMiddleware.js File Reference

Variables

- `const jwt` = `require("jsonwebtoken")`
- module `exports`

2.5.1 Variable Documentation

2.5.1.1 exports

```
module exports
```

Initial value:

```
=
(roles = []) =>
(req, res, next) => {
  const token = req.header("Authorization")?.replace("Bearer ", "");
  if (!token) return res.status(401).json({ msg: "Yetkisiz" });
  try {
    const decoded = jwt.verify(token, process.env.JWT_SECRET);
    if (roles.length && !roles.includes(decoded.role))
      return res.status(403).json({ msg: "Erişim yok" });
    req.user = decoded;
    next();
  } catch {
    res.status(401).json({ msg: "Token geçersiz" });
  }
}
```

2.5.1.2 jwt

```
const jwt = require("jsonwebtoken")
```

2.6 src/models/Level.js File Reference

Variables

- `const mongoose` = `require("mongoose")`
- `const LevelSchema`
- module `exports` = `mongoose.model("Level", LevelSchema)`

2.6.1 Variable Documentation

2.6.1.1 exports

```
module exports = mongoose.model("Level", LevelSchema)
```

2.6.1.2 LevelSchema

```
const LevelSchema
```

Initial value:

```
= new mongoose.Schema({  
  name: String,  
  permissions: [String],  
})
```

2.6.1.3 mongoose

```
const mongoose = require("mongoose")
```

2.7 src/models/Organizer.js File Reference

Variables

- `const mongoose` = `require("mongoose")`
- `const OrganizerSchema`
- `module exports` = `mongoose.model("Organizer", OrganizerSchema)`

2.7.1 Variable Documentation

2.7.1.1 exports

```
module exports = mongoose.model("Organizer", OrganizerSchema)
```

2.7.1.2 mongoose

```
const mongoose = require("mongoose")
```

2.7.1.3 OrganizerSchema

```
const OrganizerSchema
```

Initial value:

```
= new mongoose.Schema(  
  {  
    user: { type: mongoose.Schema.Types.ObjectId, ref: "User", required: true },  
    company: { type: String, required: true },  
    address: String,  
    phone: String,  
  },  
  { timestamps: true }  
)
```

2.8 src/models/Profile.js File Reference

Variables

- `const mongoose` = `require("mongoose")`
- `const ProfileSchema`
- module `exports` = `mongoose.model("Profile", ProfileSchema)`

2.8.1 Variable Documentation

2.8.1.1 exports

```
module exports = mongoose.model("Profile", ProfileSchema)
```

2.8.1.2 mongoose

```
const mongoose = require("mongoose")
```

2.8.1.3 ProfileSchema

```
const ProfileSchema
```

Initial value:

```
= new mongoose.Schema({  
  firstName: String,  
  lastName: String,  
  phone: String,  
  avatarUrl: String,  
})
```

2.9 src/models/User.js File Reference

Variables

- `const mongoose` = `require("mongoose")`
- `const UserSchema`
- module `exports` = `mongoose.model("User", UserSchema)`

2.9.1 Variable Documentation

2.9.1.1 exports

```
module exports = mongoose.model("User", UserSchema)
```


2.9.1.2 mongoose

```
const mongoose = require("mongoose")
```

2.9.1.3 UserSchema

```
const UserSchema
```

Initial value:

```
= new mongoose.Schema(  
  {  
    email: { type: String, required: true, unique: true },  
    password: { type: String, required: true },  
    role: {  
      type: String,  
      enum: ["admin", "organizer", "customer"],  
      default: "organizer",  
    },  
    profile: { type: mongoose.Schema.Types.ObjectId, ref: "Profile" },  
  },  
  { timestamps: true }  
)
```

2.10 src/routes/authRoutes.js File Reference

Variables

- `const router = require("express").Router()`
- `const { login, resetPassword, forgotPassword } = require("../controllers/authController")`
- `module exports = router`

2.10.1 Variable Documentation

2.10.1.1 const

```
const { login, resetPassword, forgotPassword } = require("../controllers/authController")
```

2.10.1.2 exports

```
module exports = router
```

2.10.1.3 router

```
const router = require("express").Router()
```

2.11 src/routes/organizerRoutes.js File Reference

Variables

- `const router` = `require("express").Router()`
- `const`
- `const auth` = `require("../middlewares/authMiddleware")`
- module `exports` = `router`

2.11.1 Variable Documentation

2.11.1.1 auth

```
const auth = require("../middlewares/authMiddleware")
```

2.11.1.2 const

```
const
```

Initial value:

```
{
  createOrganizer,
  getAllOrganizers,
  updateOrganizer,
  deleteOrganizer,
} = require("../controllers/organizerController")
```

2.11.1.3 exports

```
module exports = router
```

2.11.1.4 router

```
const router = require("express").Router()
```

2.12 src/utils/emailService.js File Reference

Functions

- `if (!apiKey || !apiKey.startsWith("SG."))`
- function async `sendWelcomeEmail` (to, tempPassword)

Variables

- `const sgMail` = `require("@sendgrid/mail")`
- `const apiKey` = `process.env.SENDGRID_API_KEY`
- `else`
- `module exports` = { `sendWelcomeEmail` }

2.12.1 Function Documentation

2.12.1.1 if()

```
if (
    !apiKey||!apiKey.  startsWith"SG.")
```

2.12.1.2 sendWelcomeEmail()

```
function async sendWelcomeEmail (
    to,
    tempPassword)
```

2.12.2 Variable Documentation

2.12.2.1 apiKey

```
const apiKey = process.env.SENDGRID_API_KEY
```

2.12.2.2 else

```
else
```

Initial value:

```
{
    sgMail.setApiKey(apiKey)
```

2.12.2.3 exports

```
module exports = { sendWelcomeEmail }
```

2.12.2.4 sgMail

```
const sgMail = require("@sendgrid/mail")
```


Index

- apiKey
 - emailService.js, [13](#)
- app
 - index.js, [7](#)
- auth
 - organizerRoutes.js, [12](#)
- authController.js
 - bcrypt, [4](#)
 - const, [4](#)
 - crypto, [4](#)
 - forgotPassword, [4](#)
 - jwt, [4](#)
 - login, [4](#)
 - resetPassword, [5](#)
 - resetTokens, [5](#)
 - User, [5](#)
- authMiddleware.js
 - exports, [8](#)
 - jwt, [8](#)
- authRoutes
 - index.js, [7](#)
- authRoutes.js
 - const, [11](#)
 - exports, [11](#)
 - router, [11](#)
- bcrypt
 - authController.js, [4](#)
 - organizerController.js, [6](#)
- connectDB
 - db.js, [3](#)
 - index.js, [7](#)
- const
 - authController.js, [4](#)
 - authRoutes.js, [11](#)
 - organizerController.js, [6](#)
 - organizerRoutes.js, [12](#)
- cors
 - index.js, [7](#)
- createOrganizer
 - organizerController.js, [6](#)
- crypto
 - authController.js, [4](#)
- db.js
 - connectDB, [3](#)
 - exports, [3](#)
 - mongoose, [3](#)
- deleteOrganizer
 - organizerController.js, [6](#)
- else
 - emailService.js, [13](#)
- emailService.js
 - apiKey, [13](#)
 - else, [13](#)
 - exports, [13](#)
 - if, [13](#)
 - sendWelcomeEmail, [13](#)
 - sgMail, [13](#)
- exports
 - authMiddleware.js, [8](#)
 - authRoutes.js, [11](#)
 - db.js, [3](#)
 - emailService.js, [13](#)
 - Level.js, [8](#)
 - Organizer.js, [9](#)
 - organizerRoutes.js, [12](#)
 - Profile.js, [10](#)
 - User.js, [10](#)
- express
 - index.js, [7](#)
- forgotPassword
 - authController.js, [4](#)
- getAllOrganizers
 - organizerController.js, [6](#)
- if
 - emailService.js, [13](#)
- index.js
 - app, [7](#)
 - authRoutes, [7](#)
 - connectDB, [7](#)
 - cors, [7](#)
 - express, [7](#)
 - organizerRoutes, [7](#)
 - PORT, [7](#)
- jwt
 - authController.js, [4](#)
 - authMiddleware.js, [8](#)
- Level.js
 - exports, [8](#)
 - LevelSchema, [8](#)
 - mongoose, [9](#)
- LevelSchema
 - Level.js, [8](#)

- login
 - authController.js, 4
- mongoose
 - db.js, 3
 - Level.js, 9
 - Organizer.js, 9
 - Profile.js, 10
 - User.js, 10
- Organizer
 - organizerController.js, 6
- Organizer.js
 - exports, 9
 - mongoose, 9
 - OrganizerSchema, 9
- organizerController.js
 - bcrypt, 6
 - const, 6
 - createOrganizer, 6
 - deleteOrganizer, 6
 - getAllOrganizers, 6
 - Organizer, 6
 - Profile, 6
 - updateOrganizer, 6
 - User, 6
- organizerRoutes
 - index.js, 7
- organizerRoutes.js
 - auth, 12
 - const, 12
 - exports, 12
 - router, 12
- OrganizerSchema
 - Organizer.js, 9
- PORT
 - index.js, 7
- Profile
 - organizerController.js, 6
- Profile.js
 - exports, 10
 - mongoose, 10
 - ProfileSchema, 10
- ProfileSchema
 - Profile.js, 10
- resetPassword
 - authController.js, 5
- resetTokens
 - authController.js, 5
- router
 - authRoutes.js, 11
 - organizerRoutes.js, 12
- sendWelcomeEmail
 - emailService.js, 13
- sgMail
 - emailService.js, 13
- src/config/db.js, 3
- src/controllers/authController.js, 4
- src/controllers/organizerController.js, 5
- src/index.js, 7
- src/middlewares/authMiddleware.js, 8
- src/models/Level.js, 8
- src/models/Organizer.js, 9
- src/models/Profile.js, 10
- src/models/User.js, 10
- src/routes/authRoutes.js, 11
- src/routes/organizerRoutes.js, 12
- src/utils/emailService.js, 12
- updateOrganizer
 - organizerController.js, 6
- User
 - authController.js, 5
 - organizerController.js, 6
- User.js
 - exports, 10
 - mongoose, 10
 - UserSchema, 11
- UserSchema
 - User.js, 11