



UserController

```
CreateUser(userName : string, password : string, role : UserRole)
UpdateUser(id : int, userName : string, password : string, role : UserRole)
DeleteUser(id : int)

GetAllUsers()

GetUserId(userName : string)
```

ComplaintController

```
CreateComplaint(id: int, title : string, description : string, author : User, publishDate : DateTime, status : ComplaintStatus, severity : ComplaintSeverity)
UpdateComplaint(id: int, title : string, description : string, author : User, publishDate : DateTime, status : ComplaintStatus, severity : ComplaintSeverity, currentUser : User)

GetAllComplaintsByUser(user: User)
GetAllComplaintsByStatusOrSeverity(status : ComplaintStatus?, severity : ComplaintSeverity?)
GetComplaint(id : int)
```

AnnouncementController

```
CreateAnnouncement(title : string, description : string, author : User, publishDate : DateTime, isImportant : bool, isSticky : bool)
UpdateAnnouncement(id: int, title : string, description : string, author : User, publishDate : DateTime, isImportant : bool, isSticky : bool, currentUser : User)
DeleteAnnouncement(id : int, currentUser: User)
GetAnnouncementsRange(page : int, count : int)
GetAnnouncement(id : int)
AddComment(id: int, title : string, description : string, author : User, publishDate : DateTime)
UpVote(id : int, currentUser : User)
DownVote(id : int, currentUser : User)
```

CommentController

```
CreateComment(title : string, description : string, author : User, publishDate : DateTime)
UpdateComment(title : string, description : string, author : User, publishDate : DateTime, currentUser : User)
DeleteAnnouncement(id : int, currentUser : User)
GetComment(id : int)
AddResponse(id: int, title : string, description : string, author : User, publishDate : DateTime)
UpVote(id : int, currentUser : User)
DownVote(id : int, currentUser : User)
```

EventController

```
CreateEvent(id: int, title : string, description : string, author : User, publishDate : DateTime, startDate : DateTime, endDate : DateTime)
UpdateEvent(id: int, title : string, description : string, author : User, publishDate : DateTime, startDate : DateTime, endDate : DateTime, currentUser : User)
DeleteEvent(id : int, currentUser : User)
GetAllEventsByMonth(queryDate : DateTime)
GetEvent(id : int)
```