|  |  |  |
| --- | --- | --- |
| **Description** | **Method + Path**  **(path is always /api.php)** | **Request/Response** |
| Check login status (is a administrator logged in or not) | GET | **Request** {ctrl: login }  **Response** (administrator logged in) **{**  “**adminID**":2,  "**adminName**":"Judy”,  "**roleID**":4,  "**roleName**":"owner",  "**adminPhone**":"0527156769",  "**adminEmail**":[judithguttmann@gmail.com](mailto:judithguttmann@gmail.com)  }  **Response** (no admin logged in){  **“no”**  **}** |
| Login (send admin email & password to server)  If login successful admin object  Of administrator who logged in returned- else message returned that no such admin existst | GET | **Request**{  **ctrl:** login  **userEmail:** [judithguttmann@gmail.com](mailto:judithguttmann@gmail.com),  **userPassword:** wedding5771,  }  **Response** (login successful) **{**  “**adminID**":2,  "**adminName**":"Judy”,  "**roleID**":4,  "**roleName**":"owner",  "**adminPhone**":"0527156769",  "**adminEmail**":[judithguttmann@gmail.com](mailto:judithguttmann@gmail.com)  }  **Response** (login unsuccessful) **{**  "no administrator with this email and password found"  } |
| Logout – delete admin session object of server | DELETE | **Request** {**ctrl**: login }  **Response** (no admin logged in){  **“no”**  **}** |
| Get all admin roles – used to fill combo box on admin update panel | GET | **Request** {**ctrl**: role}  **Response {**  [{"roleID":5,"roleName":"manager"},  {"roleID":4,"roleName":"owner"},  {"roleID":6,"roleName":"sales"}]  } |
| Get admin by Name and Email –used for remote validation on client to ensure no other admin with same name and email already exists  adminID: -1 no such admin found  adminID: <> -1 such admin found | GET | **Request**{  **ctrl:** administrator,  **adminName:** judy,  **adminEmail:**judithguttmann@gmail.com  }  **Response {**  {"status":"ok",  "adminID":-1}  }  **Response** (no session variable on server - user no longer logged in) **{**  {"status":"no longer logged in",  "action":"GET administrator"}  } |
| Get all administrators | GET | **Request**{  **ctrl:** administrator,  }  **Response {**  [{"**adminID**":2,  "**adminName**":"judy",  "**roleID**":4,  "**roleName**":"owner",  "**adminPhone**":"0527156769",  "**adminEmail**":"a@gmail.com"},  {"**adminID**":3,  "**adminName**":"Chaim",  "**roleID**":5,  "**roleName**":"manager",  "**adminPhone**":"0559888705  “**adminEmail**":"b@gmail.com} ... ]  } |
| Create new Admin  duplicateAdmin & adminImageSize  are used for client validations (they are NOT required on the server)  They can be found in Request because they are in form and whole form is serialized with the  $("#frmCUD").serialize();  Statement. | POST | **Request**{  **ctrl:** administrator,  **adminID**: 0,  **roleName**: sales,  **roleID**: 6,  **duplicateAdmin**: ,  **adminName**: meir,  **adminPhone**: 12345,  **adminEmail**: [c@gmail.com](mailto:c@gmail.com),  **adminPassword**: gunibush,  **adminImageSize**: ,  **adminImage**: ; filename="001.JPG"  }  **Response** {  "status":"ok",  "action":"Create administrator",  "message":" administrator added successfully",  "new\_adminID":11  } |
| Update Admin  duplicateAdmin & adminImageSize  are used for client validations (they are NOT required on the server)  They can be found in Request because they are in form and whole form is serialized with the  $("#frmCUD").serialize();  Statement. | PUT | **Request**{  **ctrl:** administrator,  **adminID**: 2,  **roleName**: manager,  **roleID**: 5,  **duplicateAdmin**: ,  **adminName**: meir ilson,  **adminPhone**: 12345,  **adminEmail**: [c@gmail.com](mailto:c@gmail.com),  **adminImageSize**: ,  }  **Response** {  "status":"ok",  "action":" Update administrator",  "message":" administrator updated successfully"  } |
| Delete Admin | DELETE | **Request**{  **ctrl:** administrator,  **adminID**: 2  }  **Response** {  "status":"ok",  "action":" Delete administrator",  "message":" administrator deleted successfully"  } |
| Get admin by Name and Email –used for remote validation on client to ensure no other course with same already exists  courseID: -1 no such course found  courseID: <> -1 such course found | GET | **Request**{  **ctrl:** course,  **courseName**: c#  }  **Response {**  {"status":"ok",  " courseID":-1}  }  **Response** (no session variable on server - user no longer logged in) **{**  {"status":"no longer logged in",  "action":"GET course"}  } |
| Get all courses | GET | **Request**{  **ctrl:** course,  }  **Response {**  [{"courseID":5,  "courseName":"babysitting",  "courseDescription":"earn extra money",  "numberOfStudentsForCourse":7,  "studentIDs":"16,1,4,15,14,5,3"},  {"courseID":3,  "courseName":"c#",  "courseDescription":"learn to program- fast",  "numberOfStudentsForCourse":8,  "studentIDs":"16,1,2,4,8,15,14,7"} … ]  } |
| Create new Course  duplicateCourse & courseImageSize  are used for client validations (they are NOT required on the server)  They can be found in Request because they are in form and whole form is serialized with the  $("#frmCUD").serialize();  Statement. | POST | **Request**{  **ctrl:** course,  courseID: 0,  duplicateCourse:,  courseName: karate,  courseDescription: defend yourself,  courseImageSize:,  courseImage: ; filename="002.JPG"  }  **Response** {  "status":"ok",  "action":"Create course",  "message":" course added successfully",  "new\_courseID":15  } |
| Update Course  duplicateCourse & courseImageSize  are used for client validations (they are NOT required on the server)  They can be found in Request because they are in form and whole form is serialized with the  $("#frmCUD").serialize();  Statement. | PUT | **Request**{  **ctrl:** course,  courseID: 3,  duplicateCourse:,  courseName: c# - beginners,  courseDescription: learn to program- fast,  courseImageSize:  }  **Response** {  "status":"ok",  "action":" Update course",  "message":" course updated successfully"  } |
| Delete Course | DELETE | **Request**{  **ctrl:** course,  **courseID**: 3  }  **Response** {  "status":"ok",  "action":" Delete course ",  "message":" course deleted successfully"  } |
|  |  |  |
|  |  |  |
|  |  |  |