

Die Höhere Fachschule für Technik Mittelland - eine Klasse für sich. Als führende Schule mit starker regionaler Verankerung in der Präzisionsindustrie unterstützen wir technische Berufsleute auf ihrem Karriereweg. Rund 100 Dozierende vermitteln über 450 Studierenden praxisorientierte Kompetenzen in den aktuellen Trends wie Industrie 4.0, Digitalisierung, erneuerbare Energien und zukunftsweisenden Fertigungstechnologien. Für die Unterstützung in der IT suchen wir per 1. August 2022 oder nach Vereinbarung einen/eine

Praktikant*in “System Administrator” 60-100%

Ihre Aufgaben

Zu Ihren Aufgaben gehört die Konzeption, Verwaltung und Wartung der IT-Infrastruktur, die Pflege der Systeme Linux und Windows sowie von vielen weiteren Umsystemen. Die Installation und Konfiguration von neuen Komponenten fallen ebenso in Ihr Aufgabengebiet. Als weitere Aufgabe sind Sie das Bindeglied zwischen der IT und der Administration bezüglich Erweiterung von Fachapplikationen. Der Support rund um die IT-Arbeitsinfrastruktur (MO-SA) und der mobilen Geräte wie Notebooks, Tablets und Smartphones runden Ihre Aufgaben ab.

Ihr Profil

Sie sind in Ausbildung mit Fachrichtung Systemtechnik oder haben diese erfolgreich abgeschlossen und kennen sich mit Windows, Linux, Netztopologien und hybriden Kommunikationslösungen bestens aus. Sie sind eine offene, interessierte und kommunikative Persönlichkeit. Sie wollen Ihren Kunden ein exzellentes Benutzererlebnis bieten und setzen sich für eine reibungslos funktionierende IT-Infrastruktur ein. Die Zufriedenheit der Kunden steht im Zentrum Ihres Handelns.

Wir bieten

Wir bieten ein lehrreiches, dynamisches und innovatives Arbeitsumfeld, ein motiviertes Team, zeitgemässe Entlöhnung und flexible Arbeitszeiten an zwei top modernen Standorten.

Interessiert? Dann senden Sie Ihre vollständigen Bewerbungsunterlagen per E-Mail an: bruno.borer@hftm.ch

Für telefonische Auskünfte steht Ihnen Herr Bruno E. Borer, Leiter Informatik & technische Dienste der hftm, gerne zur Verfügung, Tel. +41 32 654 12 00.

kompetent. clever. agil.

HAUPTSITZ/ RECHNUNGSADRESSE

Höhere Fachschule Technik Mittelland AG
Sportstrasse 2 · CH-2540 Grenchen
T +41 32 654 12 00
sekretariat@hftm.ch · www.hftm.ch

STANDORT BIEL/ BIENNE

Ecole supérieure technique Mittelland SA
Rue d'Aarberg 46 · CH-2503 Bienne
T +41 32 654 12 02

SOCIAL MEDIA



```
private ServiceLocator serviceLocator;
public TournamentService() {
    serviceLocator = ServiceLocator.getInstance();
}
public GameDurationDTO getShortestGameDuration() {
    GameDurationDTO responseObject = null;
    Map<String, String> queryParams = new HashMap<>();
    queryParams.put("name", "T");
    queryParams.put("key", "T");
    Invocation.Builder builder = serviceLocator.getInvocationBuilder();
    Response response = builder.get(GameDurationDTO.class);
    String s = response.readEntity(String.class);
    ObjectMapper mapper = new ObjectMapper();
    try {
        responseObject = mapper.readValue(s, GameDurationDTO.class);
    } catch (Exception ex) {
        ex.printStackTrace();
    }
    return responseObject;
}
public TournamentStandingsDTO getStandings() {
    TournamentStandingsDTO responseObject = null;
    Map<String, String> queryParams = new HashMap<>();
    queryParams.put("name", "T");
    queryParams.put("key", "T");
    Invocation.Builder builder = serviceLocator.getInvocationBuilder();
    Response response = builder.get(TournamentStandingsDTO.class);
    String s = response.readEntity(String.class);
    ObjectMapper mapper = new ObjectMapper();
    try {
        responseObject = mapper.readValue(s, TournamentStandingsDTO.class);
    } catch (Exception ex) {
        ex.printStackTrace();
    }
    return responseObject;
}
public GoalDifferenceDTO getBestGoalDifference() {
    GoalDifferenceDTO responseObject = null;
    Map<String, String> queryParams = new HashMap<>();
    queryParams.put("name", "T");
    queryParams.put("key", "T");
    Invocation.Builder builder = serviceLocator.getInvocationBuilder();
    Response response = builder.get(GoalDifferenceDTO.class);
    String s = response.readEntity(String.class);
    ObjectMapper mapper = new ObjectMapper();
    try {
        responseObject = mapper.readValue(s, GoalDifferenceDTO.class);
    } catch (Exception ex) {
        ex.printStackTrace();
    }
    return responseObject;
}
public TeamListDTO getWaitingList() {
    TeamListDTO responseObject = null;
    Map<String, String> queryParams = new HashMap<>();
    queryParams.put("name", "T");
    queryParams.put("key", "T");
    Invocation.Builder builder = serviceLocator.getInvocationBuilder();
    Response response = builder.get(TeamListDTO.class);
    String s = response.readEntity(String.class);
    ObjectMapper mapper = new ObjectMapper();
    try {
        responseObject = mapper.readValue(s, TeamListDTO.class);
    } catch (Exception ex) {
        ex.printStackTrace();
    }
    return responseObject;
}
public LastUpdatedGameDTO getLastUpdatedGame() {
    LastUpdatedGameDTO responseObject = null;
    Map<String, String> queryParams = new HashMap<>();
    queryParams.put("name", "T");
    queryParams.put("key", "T");
    Invocation.Builder builder = serviceLocator.getInvocationBuilder();
    Response response = builder.get(LastUpdatedGameDTO.class);
    String s = response.readEntity(String.class);
    ObjectMapper mapper = new ObjectMapper();
    try {
        responseObject = mapper.readValue(s, LastUpdatedGameDTO.class);
    } catch (Exception ex) {
        ex.printStackTrace();
    }
    return responseObject;
}
public int keyCheck(String tournamentName, String key) {
    Map<String, String> queryParams = new HashMap<>();
    queryParams.put("name", "T");
    queryParams.put("key", "T");
    Invocation.Builder builder = serviceLocator.getInvocationBuilder();
    Response response = builder.get(int.class);
    return response.getStatus();
}
public TournamentListDTO getTournamentList() {
    TournamentListDTO responseObject = null;
    Map<String, String> queryParams = new HashMap<>();
    queryParams.put("tableMap", "T");
    Invocation.Builder builder = serviceLocator.getInvocationBuilder();
    Response response = builder.get(TournamentListDTO.class);
    String s = response.readEntity(String.class);
    ObjectMapper mapper = new ObjectMapper();
    try {
        responseObject = mapper.readValue(s, TournamentListDTO.class);
    } catch (Exception ex) {
        ex.printStackTrace();
    }
    return responseObject;
}
```