## 1 CinemaBLL类

### 方法:

public List<MOVIE\_SCHEDULE> GetMovies(string newCinemaName)

根据电影院名字 返回该电影院的放映电影时刻列表

public CINEMA GetCinema(string newCinemaName)

根据电影院名字 返回一个电影院对象

public List<CINAME\_COMMENT> GetLatestComment(string newCinemaName)

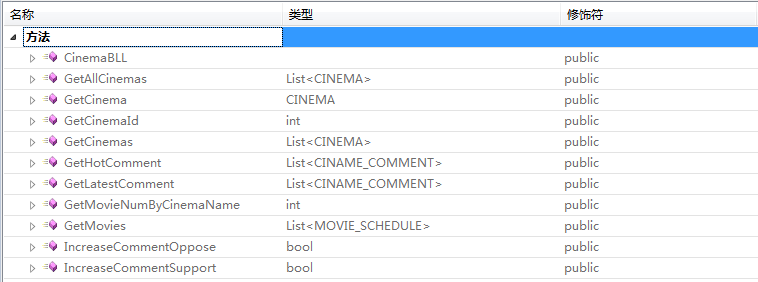
根据电影院的名字 检索最近的评论 并且展示

public List<CINAME\_COMMENT> GetHotComment(string newCinemaName)

调用检索函数 返回最热评论列表

public List<CINEMA> GetCinemas(string newAddr)

调用检索函数 返回电影院列表



## 2 MoviesBLL类

### 方法:

public MOVIE GetMovie(string newName)

返回一个电影对象

public List<MOVIE\_SCHEDULE> GetMovieScheduleByMovieName(string newMovieName)

电影的名字获取电影时刻列表

public string GetPicture(string newMovieName)

根据电影的名称获得电影的海报路径

public List<MOVIE\_COMMENT> GetHotComments(string newMovieName)

返回电影的热度评价列表。

public List<MOVIE\_COMMENT> GetLatestComments(string newMovieName)

根据新传入来的电影名字返回最新的评价列表。

public List<MOVIE> GetHotRank()

返回热度排行列表。

public List<MOVIE> GradeRank()

返回评分电影排行

public List<MOVIE> GetMoviesBySearchText(bool isOnShow, string newSearchText)

根据是否上映和电影名或导演名搜索

public string GetMoviePicture(string newMovieName)

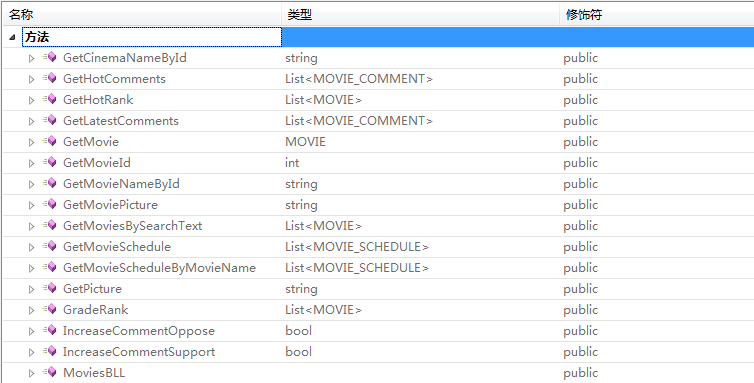
根据电影名得到电影图片的地址

public List<MOVIE\_SCHEDULE> GetMovieSchedule(string newMovieName)

根据电影名 得到电影时刻表

public string GetCinemaNameById(int newCinemaId)

根据电影院id 得到电影院名



## 3 UersBLL类

### 方法

public bool CheckPassword(string newPassword)

{

/\*

修改密码之前检查老密码是否正确

\*/

return s.SearchUser(id, newPassword);//实现时修改

}

public bool CheckPassword(string newEmail, string newPassword)

{

/\*

登录时的检查

\*/

return s.SearchUserInfo(newEmail, newPassword);//实现时修改

//返回值是false 时 提示相关信息

}

public bool UpdatePassword(string newEmail, string newPassWord)

{

//更新新密码

return U.UpdatePassword(newEmail, newPassWord);//实现时修改

}

public bool PreferenceSave(string newEmail, string newType)

{

//保存用户偏好信息

User newUser = new User();

return newUser.PreferenceSave(newEmail, newType)

}

public List<MOVIE> GetcollectionByUserID(int newId)

{

//返回用户收藏电影信息

}

public void MadeCommentOnCinema(string newCinemaName, int newId, CINAME\_COMMENT newComment)

{

//添加对电影院的评论 传入一个电影院名字和一个评论对象

}

public void MadeCommentOnMovie(string newMovieName, MOVIE\_COMMENT newComment)

{

//添加对电影的评论 传入一个电影名字和一个评论对象

}

public bool CreateUser(string newName, string newEmail, string newePassword)

{

//创建一个新用户

return flag;//实现时修改 \*\*\*\*\*\*\*\*\*\*\*\*已修改

}

//public bool CheckEmail(string newEmail)

//{

// return false;//实现时修改

//}

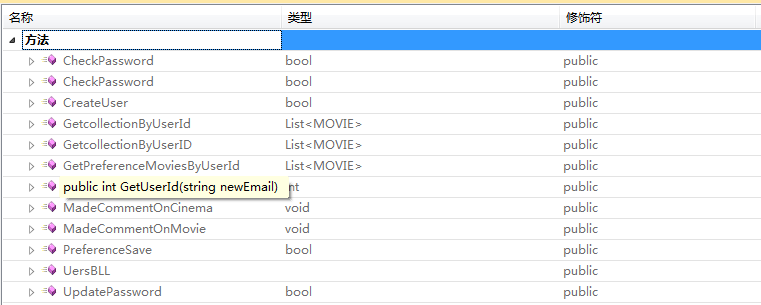
public int GetUserId(string newEmail)

{

//获取用户ID

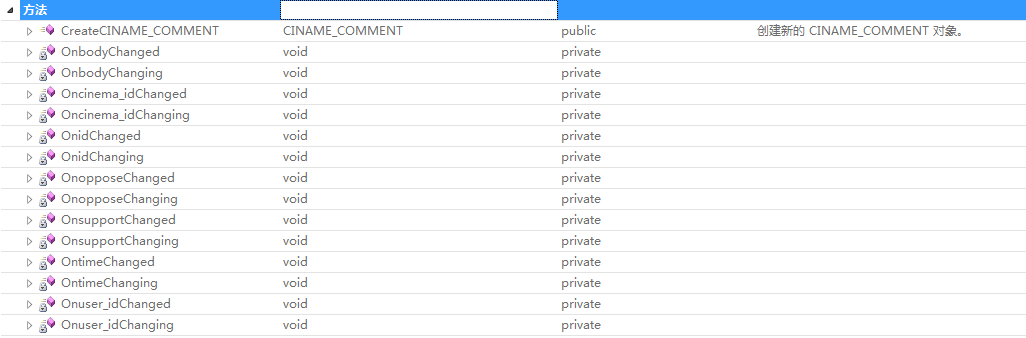
return user.GetUserId(newEmail);

}

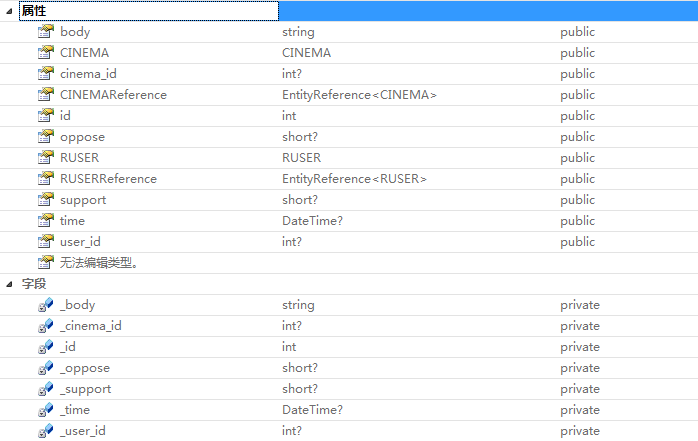


## 4 CINAME\_COMMENT类

### 方法



### 属性:



## 5 Cinema类

### 方法

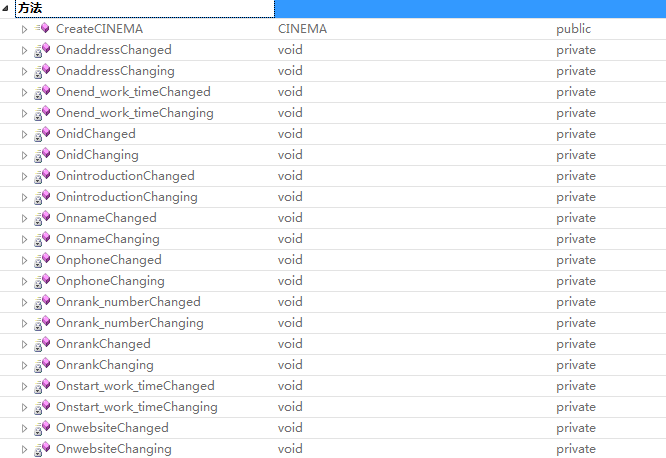


### 属性:

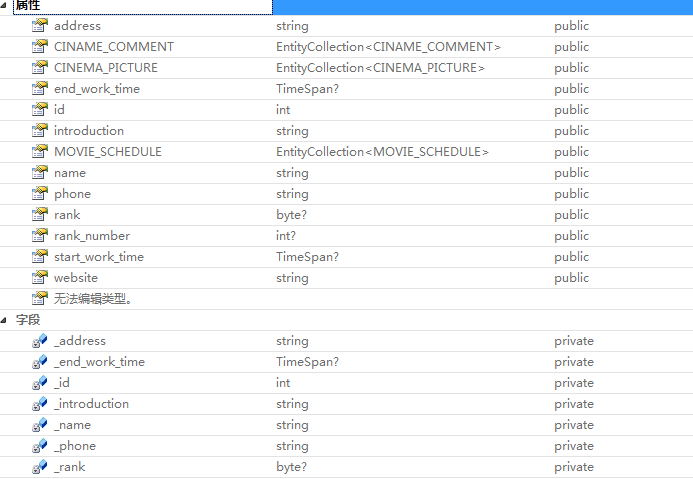


## 6 CINAME类

### 方法

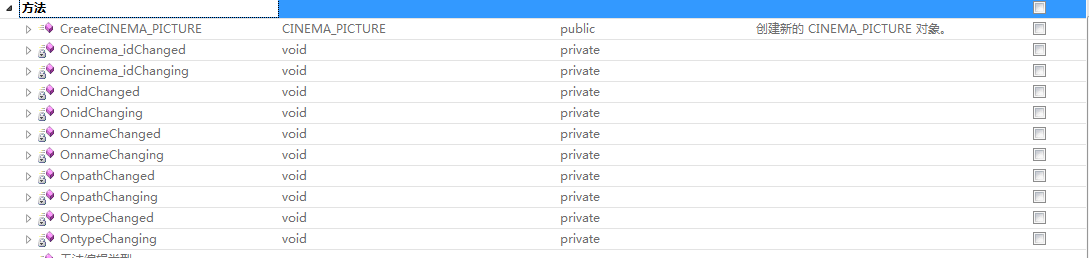


### 属性:

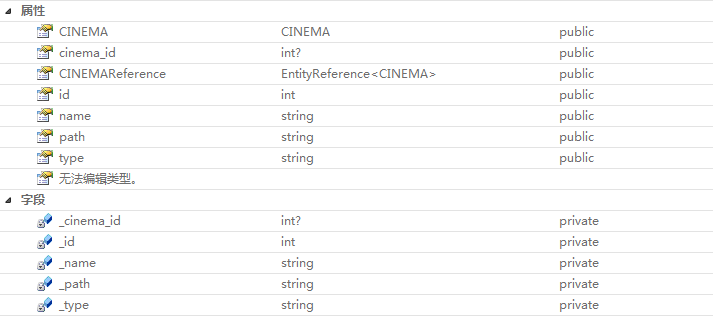


## 7 CINAME\_PICTURE类

### 方法

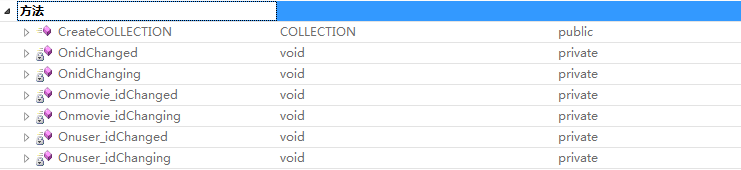


### 属性:

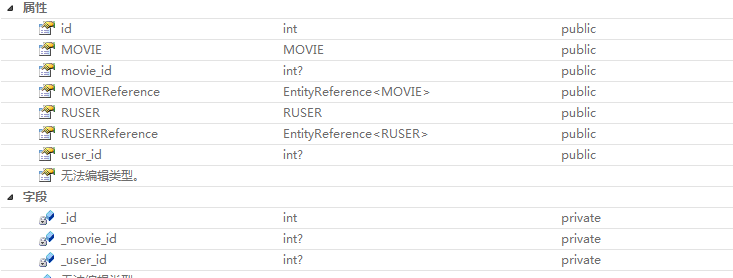


## 8 COLLECTION类

### 方法

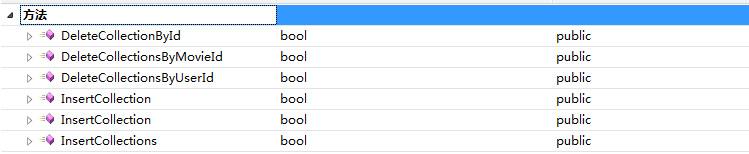


### 属性:



## 9Collections类

### 方法

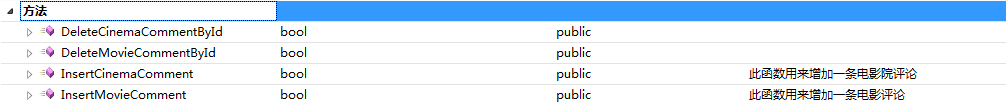


### 属性:



## 10 Comment类

### 方法

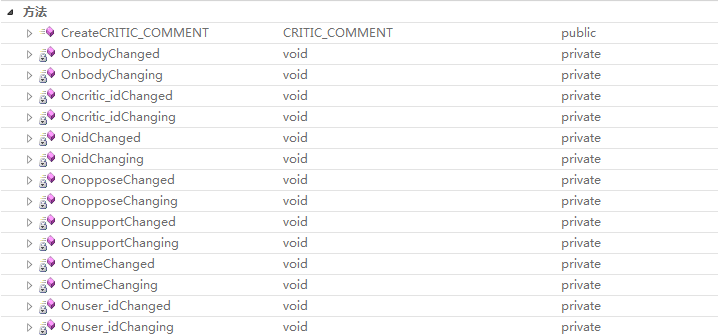


### 属性:

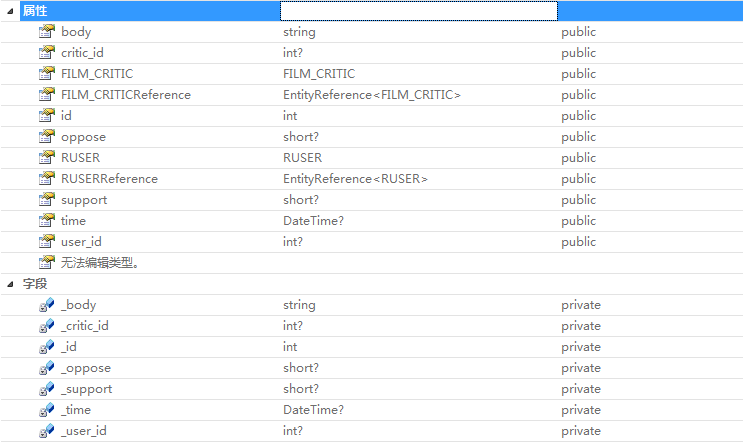


## 11 CRICT\_COMMENT类

### 方法

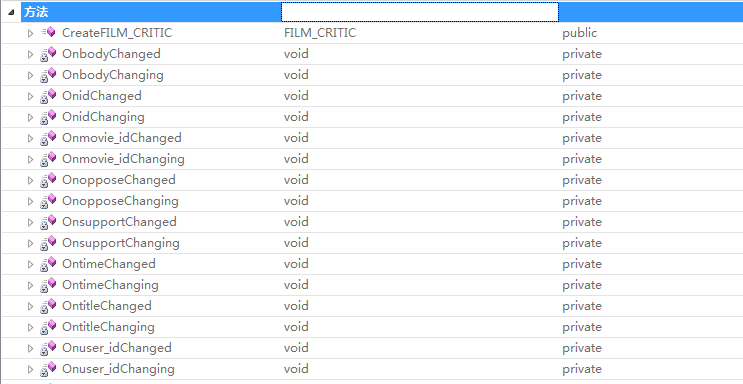


### 属性:

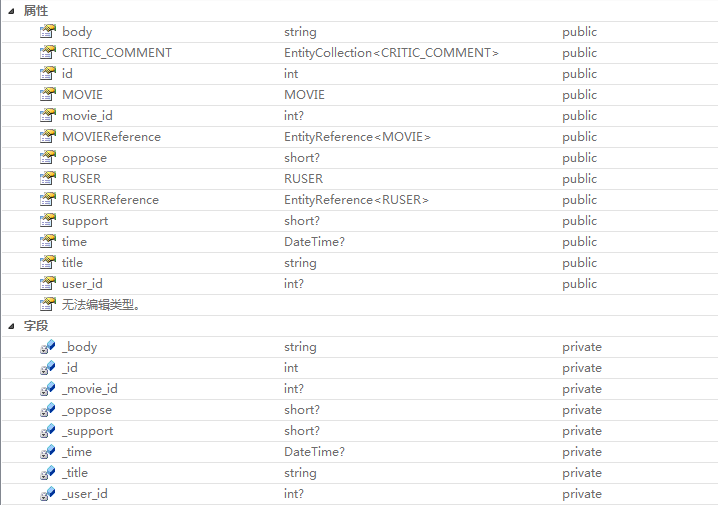


## 12 FILM\_CRITIC类

### 方法



### 属性:



## 13 Moive类

### 方法

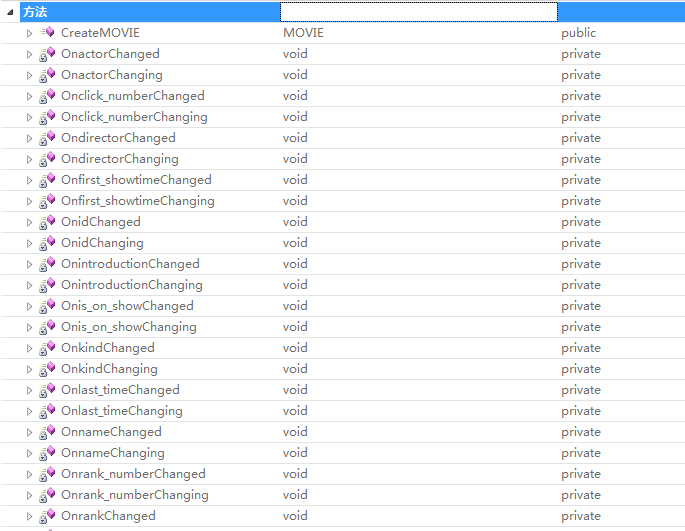


### 属性:

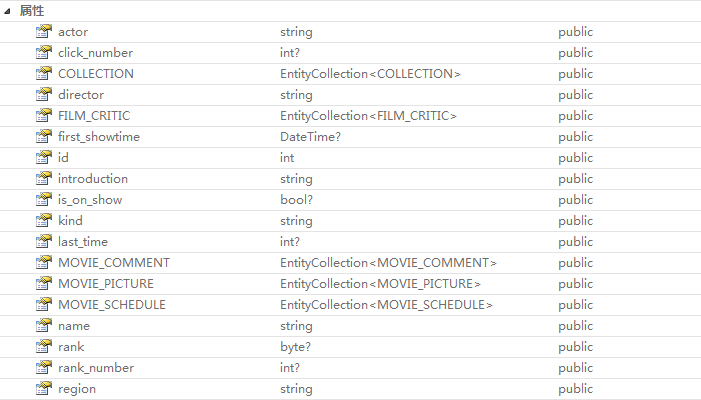


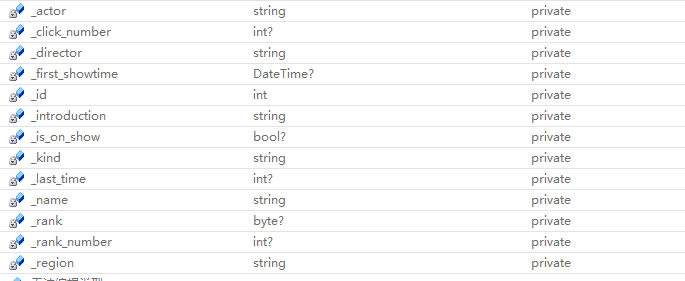
## 14 MOVIE类

### 方法



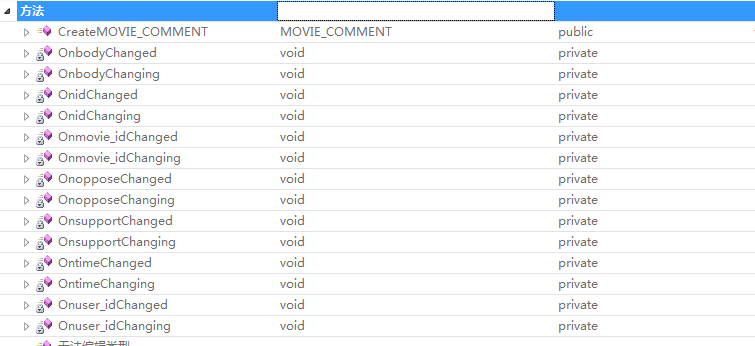
### 属性:





## 15 MOVIE\_COMMENT类

### 方法

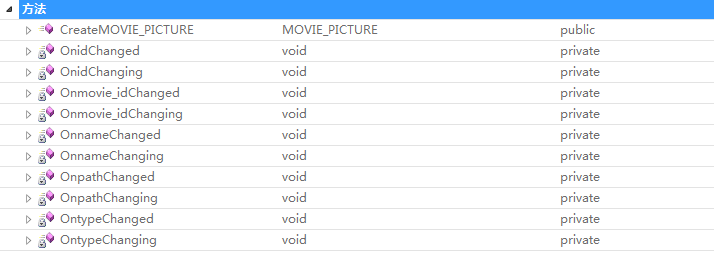


### 属性:

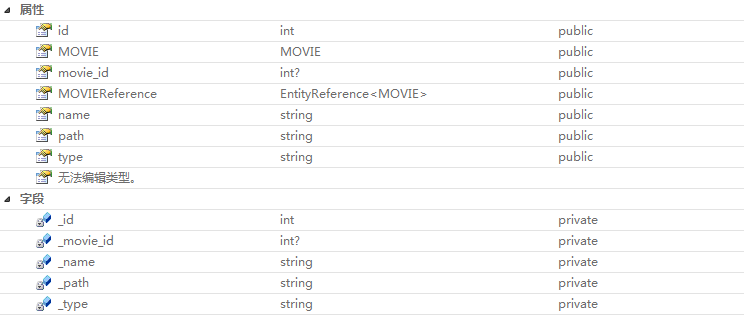


## 16 MOVIE\_PICTURE类

### 方法

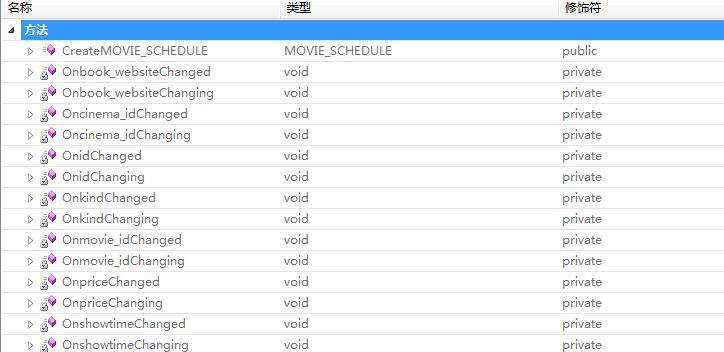


### 属性:



## 17 MOVIE\_SECHDULE类

### 方法

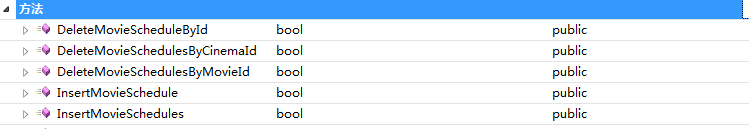


### 属性:



## 18 Movie\_schedule类

### 方法

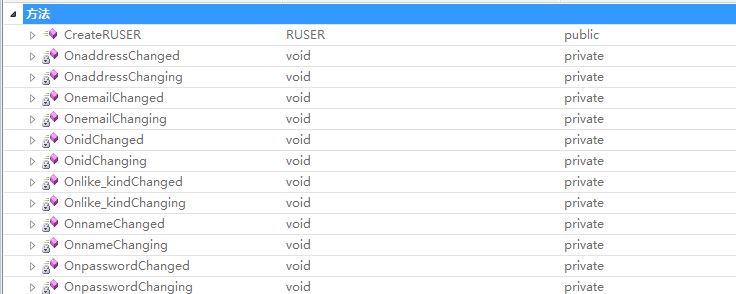


### 属性:

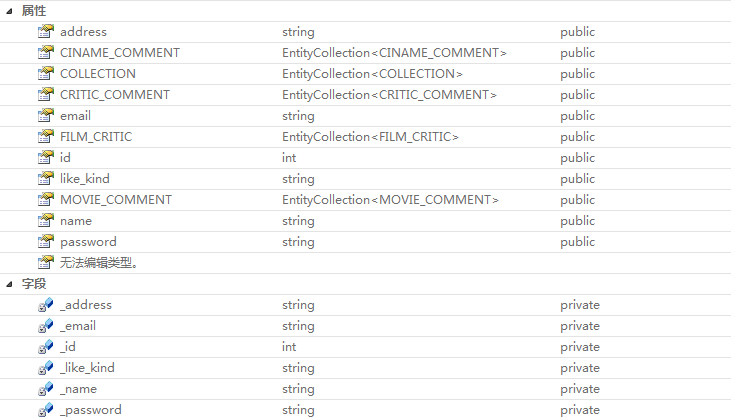


## 19 RUSER类

### 方法



### 属性:



## 20 Search类

### 方法

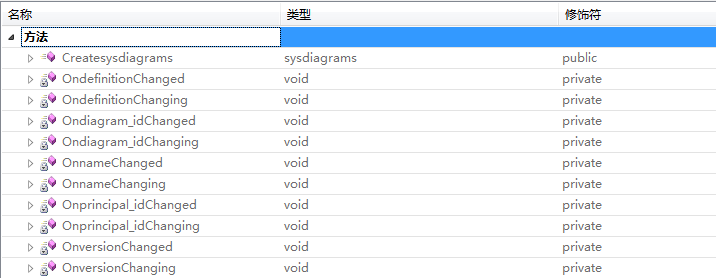


### 属性:

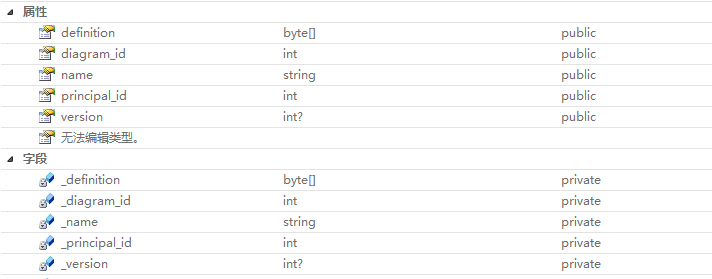


## 21 sysdiagnose类

### 方法

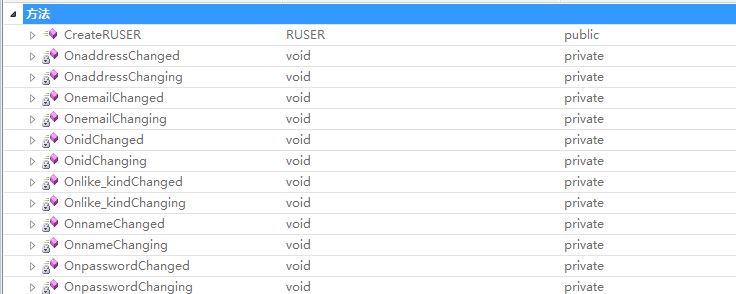


### 属性:



## 22 User类

### 方法

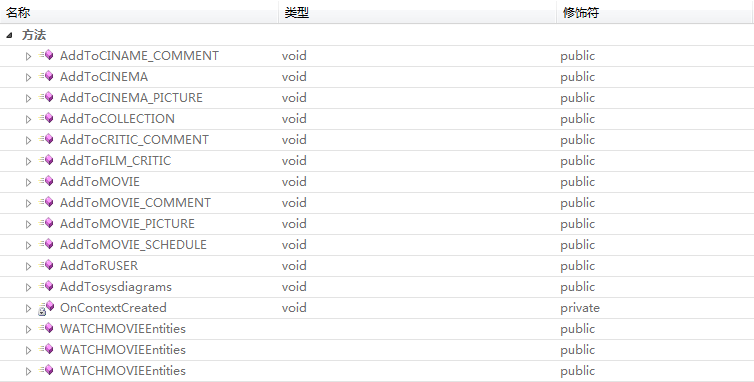


### 属性:



## 23 WATCHMOVIEEntities类

### 方法



### 属性:

