

Connection Repository

Connection repository integrates with the spring data repository. This enables your application to lookup based on either the destiny or the originating DataEntry.

Definition

```
@Repository
public interface AuthoredConnectionRepository
    extends NDBConnRepository<AuthoredConnection, String, BookDE, AuthorDE>{

}
```

This Repository definition contains not only the Connection (AuthoredConnection) but also the originating DataEntry (AuthorDE) and the destination DataEntry (BookDE).

Usage

```
@RestController
public class AnimeController{

    @Autowired
    AuthorDataRepository authorRepository;

    @Autowired
    AuthoredConnectionRepository authoredConnectionRepository;

    @GetMapping("/authored/{name}")
    public Set<AuthoredConnection> getByName(@PathVariable String name){
        Optional<AuthorDE> byName = authorRepository.findByName(id);
        if(byName.isPresent()){
            return authoredConnectionRepository.getByFrom(byName.get());
        }
    }
}
```

```

    return null;
}
@DeleteMapping("/authored/{name}")
public void deleteByName(@PathVariable String name){
    Optional<AuthorDE> byName = authorRepository.findByName(id);
    if(byName.isPresent()){
        authoredConnectionRepository.getByFrom(byName.get()).foreach(authored->
        authoredConnectionRepository.delete(authored));
    }
    return null;
}
@UpdateMapping("/authored/{name}/{country}")
public void updateCountryByName(@PathVariable String name, @PathVariable String country){
    Optional<AuthorDE> byName = authorRepository.findByName(id);
    if(byName.isPresent()){
        authoredConnectionRepository.getByFrom(byName.get()).foreach(authored->{
            authored.setCountry(country);
            authoredConnectionRepository.save(authored);
        });
    }
    return null;
}
}
}

```

Revision #5

Created 23 December 2023 19:10:53 by Nathaniel

Updated 23 December 2023 21:42:24 by Nathaniel